A Method for Aircraft Concept Exploration Using Multicriteria Interactive Genetic Algorithms

A Thesis
Presented to
The Academic Faculty

by

Michael Alexander Buonanno

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Aerospace Engineering
Georgia Institute of Technology
December 2005

A Method for Aircraft Concept Exploration Using Multicriteria Interactive Genetic Algorithms

Approved by:

Dr. Dimitri N. Mavris, Advisor
College of Engineering
Georgia Institute of Technology

Dr. David Aronstein
Raytheon Aircraft Company

Mr. Craig L. Nickol
National Air and Space Administration

Dr. Daniel P. Schrage
College of Engineering
Georgia Institute of Technology

Dr. Alan W. Wilhite
College of Engineering
Georgia Institute of Technology

Date Approved: 28 November 2005

ACKNOWLEDGEMENTS

I would like to thank my advisor Dr. Dimitri Mavris for his support from day one and for convincing me to pursue my Ph.D. Thanks to my committee members for your support, feedback, and time. I am also very appreciative of the support given to me by my GSRP technical advisors Bob McKinley, Bill Kimmel, and Craig Nickol.

My fellow graduate students have also greatly helped me during the course of my graduate career. Specifically, I'd like to thank Nick Borer, Sriram Rallabhandi, Reid Thomas, Rob McDonald, Jack Zentner, and many others for their suggestions and assistance. Finally, thanks to my friends and loved ones: Mom, Papa, Kristin, my siblings, and the countless friends I have made here at Georgia Tech. Without your encouragement I could never have gotten here.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The problem of aircraft concept selection has become increasingly difficult in recent years due to changes in the primary evaluation criteria of concepts. In the past, performance was often the primary discriminator whereas modern programs have placed increased emphasis on factors such as environmental impact, economics, supportability, aesthetics, and other metrics. The revolutionary nature of the vehicles required to simultaneously meet these conflicting requirements has prompted a shift from design using historical data regression techniques for metric prediction to the use of sophisticated physics-based analysis tools that are capable of analyzing designs outside of the historical database. The use of optimization methods with these physics-based tools, however, has proven difficult because of the tendency of optimizers to exploit assumptions present in the models and drive the design towards a solution which, while promising to the computer, may be infeasible due to factors not considered by the computer codes. In addition to this difficulty, the number of discrete options available at this stage may be unmanageable due to the combinatorial nature of the concept selection problem, leading the analyst to select a sub-optimum baseline vehicle. Some extremely important concept decisions, such as the type of control surface arrangement to use, are frequently made without sufficient understanding of their impact on the important system metrics due to a lack of historical

guidance, computational resources, or analysis tools.

This thesis discusses the difficulties associated with revolutionary system design, and introduces several new techniques designed to remedy them. First, an interactive design method has been developed that allows the designer to provide feedback to a numerical optimization algorithm during runtime, thereby preventing the optimizer from exploiting weaknesses in the analytical model. This method can be used to account for subjective criteria, or as a crude measure of un-modeled quantitative criteria. Other contributions of the work include a modified Structured Genetic Algorithm that enables the efficient search of large combinatorial design hierarchies and an improved multi-objective optimization procedure that can effectively optimize several objectives simultaneously. A new conceptual design method has been created by drawing upon each of these new capabilities and aspects of more traditional design methods.

The ability of this new technique to assist in the design of revolutionary vehicles has been demonstrated using a problem of contemporary interest: the concept exploration of a supersonic business jet. This problem was found to be a good demonstration case because of its novelty and unique requirements, and the results of this proof of concept exercise indicate that the new method is effective at providing additional insight into the relationship between a vehicle's requirements and its favorable attributes.

CHAPTER I

INTRODUCTION

A large number of authors have presented and discussed models of the engineering design process. Though each of these descriptions differ in details, all divide the design process into three major stages: conceptual, preliminary, and detailed. The conceptual stage of the design process in particular has been the subject of a great deal of research because of the large impact of decisions made during this phase on the design effort's outcome.

1.1 The conceptual design process

In [127], an ideal model of the aircraft conceptual design process is presented in a sketch called the "Design Wheel", Figure 1. In this diagram, the generation of the design concept, or the famous "back of the envelope" drawing, is depicted as an iterative process with multiple feedback loops in which the requirements and engineering analysis are used to refine the configuration.

In reality, several of the feedback loops shown in this figure have frequently been omitted for various reasons and the design process has progressed in a more serial fashion. For nearly all problems, the design process begins with requirements specification, where questions such as "how fast," "how far," and "how much" are asked and then answered through market research or a Request for Proposal (RFP). The

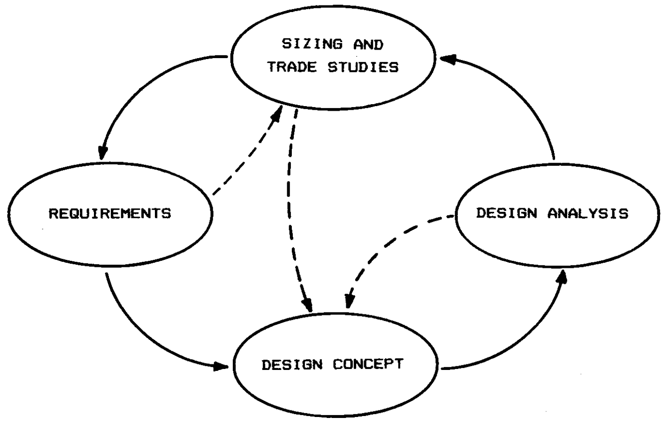


Figure 1: The Design Wheel [127]

process of synthesis follows, in which the engineers use the requirements to determine the type of vehicle concept that would be most appropriate and consider questions such as which technologies should be included in the design or what type of longitudinal control system should be used. Once a concept has been selected, the vehicle undergoes a process known as sizing, in which the geometry and weight are scaled in an iterative process until the vehicle is capable of performing the design mission. Historical information is typically used to place major subsystems and generate a rough layout. Finally, trade studies are used to optimize the configuration by generating carpet plots and examining the impact of the vehicle's characteristics on its performance and weight. The final product of this conceptual design process is a reference concept that will be subsequently refined during the preliminary and detailed stages of the design process. This process is represented graphically in Figure 2.

The importance of the conceptual design phase and the need for a more rigorous conceptual design method when designing revolutionary systems is apparent upon examination of the "Knowledge-Cost-Freedom Curve", Figure 3. This diagram depicts

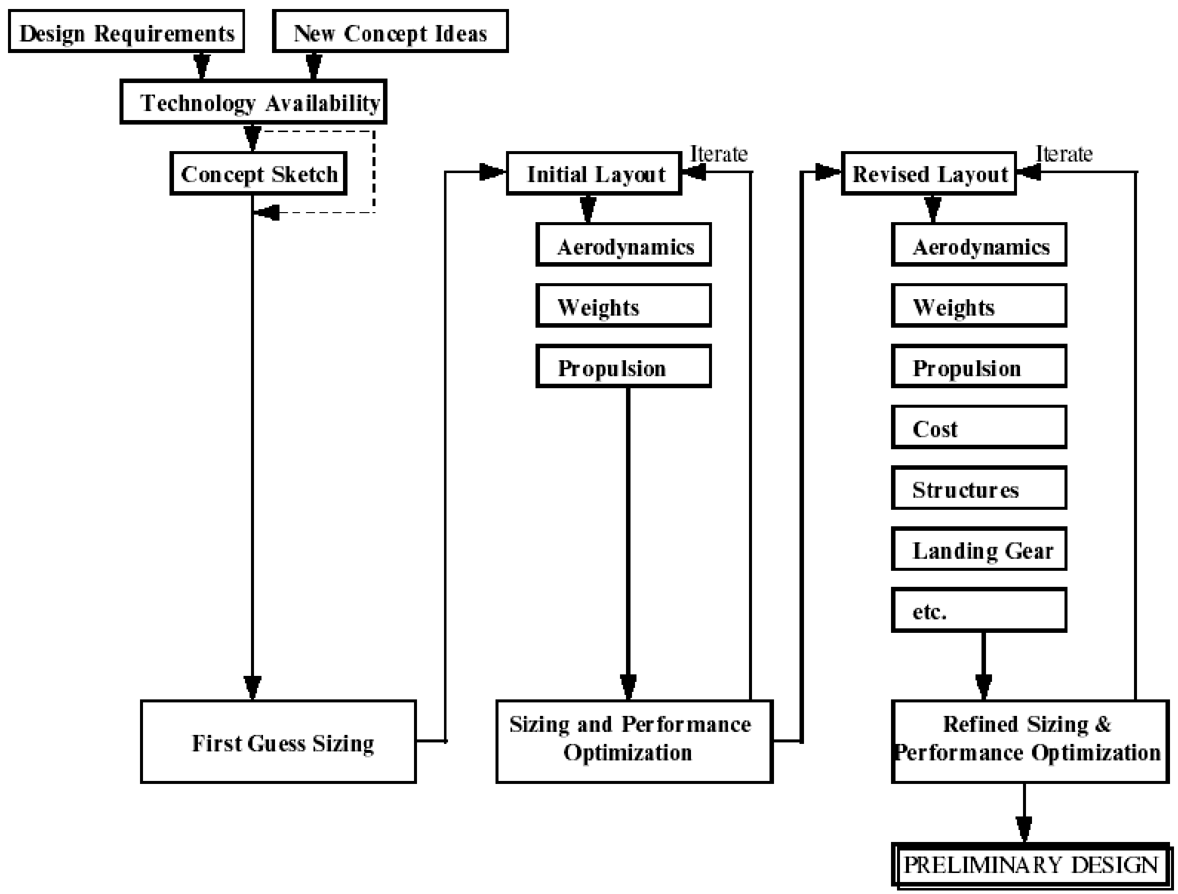


Figure 2: The aircraft conceptual design process [127]

a notional representation of the designer's relative amount of knowledge about the design under study, the amount of freedom the designer has to modify the system, and the cost committed, which are all functions of time in the design process.

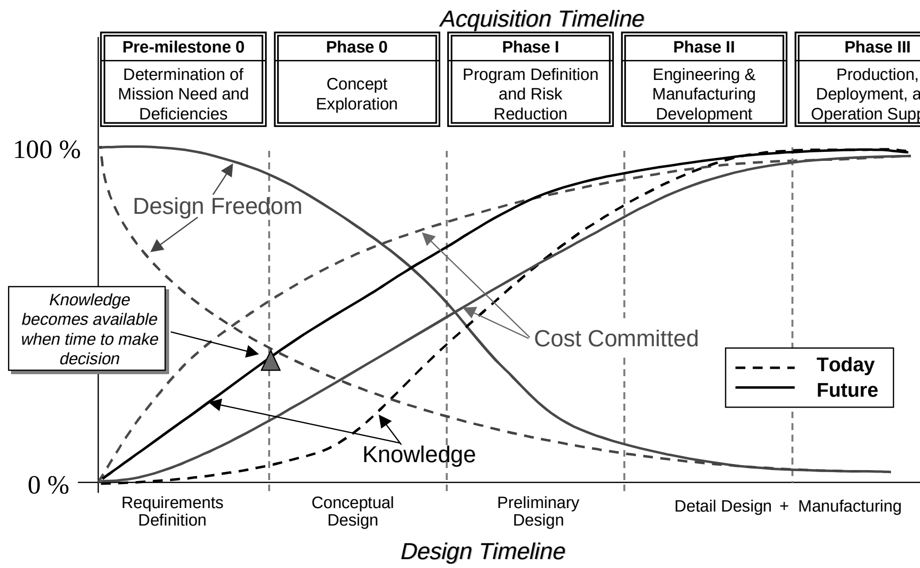


Figure 3: Distribution of knowledge, cost committed, and freedom in the design cycle [103]

The figure also suggests that the decisions made during the earliest stages of the design process are typically made with very little hard information despite the fact that they have an enormous impact on the design effort's outcome. Kidwell notes:

> The reason for the importance of the conceptual design phase is that decisions and analyses made at that time affect the entire design cycle of the aircraft; moreover, if an inherent flaw exists when the design emerges from the conceptual stage, and if it is discovered later on in the design or analysis, its solution may, because of subsystem coupling, undo much of the work already done. [83]

1.2 Motivation

The thousands of airplanes in the sky at this very moment are evidence that the traditional conceptual design process has worked reasonably well to date. However, the aircraft concept selection problem has become increasingly difficult in recent years because of increased emphasis on non-traditional metrics such as environmental effects, economics, and aesthetics. Designers are therefore forced to consider how these parameters interact with the system during the earliest stages of the program. The design of revolutionary vehicles such as commercial supersonic transports promises to be especially challenging because of possible future regulations governing sonic boom loudness and cruise emissions. [21]

The increase in problem complexity is not limited to commercial designs, and most recent military aircraft projects like the Joint Strike Fighter and F/A-18 E/F have emphasized multi-role capability rather than single-mission dominance. This requirement to be able to fly inherently conflicting missions such as both supersonic intercept and low altitude strike can make it very difficult for the designers to produce a feasible concept. [16] For both civil and military programs, these revolutionary requirements force designers to consider novel concepts that differ greatly from previous aerospace vehicles because traditional designs are simply not capable of meeting all project goals and constraints. This type of exercise, known as *creative design*, is much more challenging than a *routine design* problem, in which the attributes and methods required are well known. [43]

One of the main challenges associated with creative design problems is that the designers can no longer easily leverage historical data and past design experience during the decision-making process. When dealing with routine design problems, the engineers may already be familiar with similar design problems, and will typically have access to references that contain a large amount of historical information about previous designs. These references provide guidance via "rules of thumb" that suggest appropriate alternatives as a function of the vehicle's requirements. An example of this guidance is given in Figure 4, which depicts feasible propulsion system alternatives as a function of operating Mach number. Through the use of this information, designers can quickly formulate an appropriate design concept. For example, if one was asked to design a new 50 passenger regional jet, it would be relatively easy to find references that suggest that the best configuration is likely an airplane with a low-mounted, moderately swept wing; fuselage-mounted turbofan engines; and a T-tail. [127] [88]

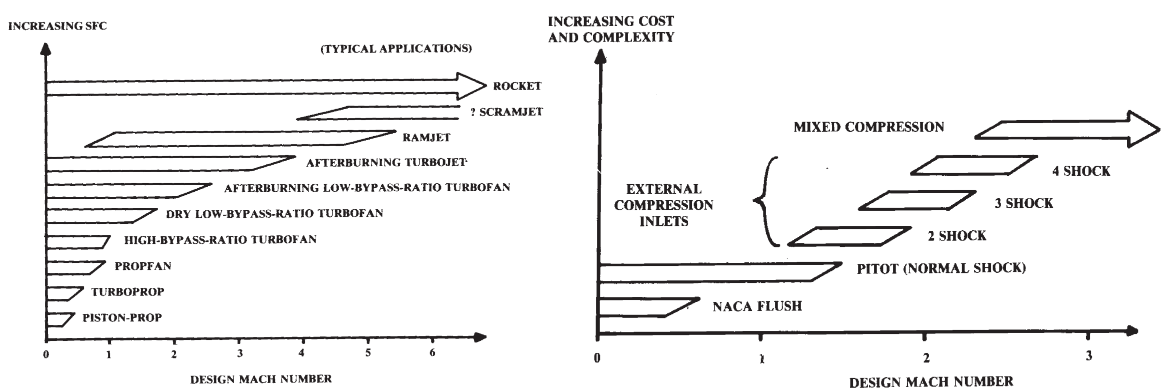


Figure 4: "Rule of Thumb" example charts [127]

Unfortunately, similar guidance is typically not available for revolutionary system design problems, and the designer may be forced to extrapolate rather than interpolate from available data or to make configuration and requirements decisions based upon intuition alone. This lack of information can lead to incorrect decisions early on and large cost increases or failure later in the project, as was the case in several real world aerospace design programs.

In the case of the United States' unsuccessful Supersonic Transport (SST) effort of the 1960s, two contributing factors to the program's ultimate demise were a failure to understand the impact of requirements on the system and the inability to correctly model system performance during concept formulation. The decision-makers chose to require the SST to fly at a cruise Mach number of 2.7, necessitating the use of titanium and steel for the structure rather than lighter and cheaper aluminum. The high cruise Mach number requirement also led Boeing to adopt a variable geometry planform that had good low- and high- speed performance, but the company's initial weight and aeroelastic estimates proved incorrect. After an attempt to correct the problem by relocating the engines to the tail, Boeing was ultimately forced to abandon the variable sweep configuration and adopt a more conventional arrow wing, as depicted in Figure 5. At this point millions of dollars had already been committed, and the configuration redefinition proved too costly: Congress cancelled the SST program in 1971 as a result of budget overruns and environmental concerns. [4]

The U.S. space shuttle program is another example that demonstrates the importance of requirements and available technologies on concept selection. Initially, NASA requirements led spacecraft designer Max Faget to design a two- stage fully reusable launch vehicle that would be capable of carrying a 25,000 lb payload to orbit from the Kennedy Space Center. (Figure 6) At the Nixon administration's insistence, NASA was forced to accept much more demanding Air Force requirements that dictated the

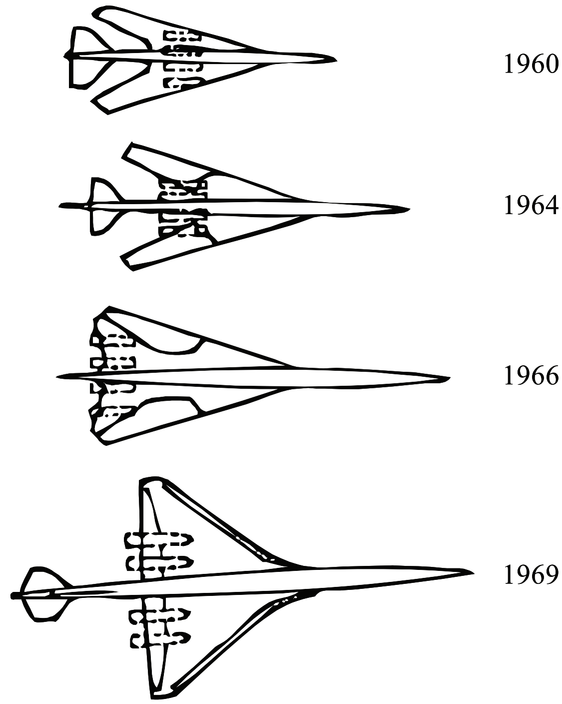


Figure 5: Evolution of the SST Design Concept [4]

ability to launch 40,000 lb payload to polar orbit from Vandenburg Air Force Base with a 1,100 nm crossrange capability. This crossrange requirement forced a concept redefinition because Faget's design using small, straight wings could only fly below 40,000 ft, limiting the vehicle's crossrange capability to about 250 nm.

At first, it seemed the Air Force requirements would be impossible to meet because the thermal protection system required to withstand prolonged heating at hypersonic speeds would weigh far too much. However, the introduction of a new silica-based insulation that was several times lighter per square foot than that used on Mercury enabled the use of delta wings with a conventional aluminum structure that could fly hypersonically and therefore meet the crossrange requirement. This delta wing configuration was eventually selected as the final design concept, but resulted in a

Figure 6: Faget's Space Transportation System Concept [66]

much heavier and more expensive vehicle. Ironically, the Air Force never used the shuttle to perform its design mission or Vandenburg AFB as a shuttle launch platform, so if the original requirements had been maintained, billions of dollars could likely have been saved over the life of the program. [66]

This problem of insufficient knowledge during the vehicle synthesis process is exacerbated by the social and psychological aspects of engineering design. Research has shown that design engineers tend to latch onto concepts: once a vehicle has been selected for in-depth analysis, the engineer tends to become attached to it and may spend a large amount of effort attempting to rework the design rather than examining other alternatives when an obstacle is encountered. [105] This tendency means that it is critical to examine as many alternatives as possible early in the design cycle, before excessive time and effort are invested in a single concept.

1.3 Challenges associated with creative design problems

Many researchers and engineers have recognized deficiencies in the traditional conceptual design process and developed a wide variety of methods designed to improve upon it. Most of these efforts have focused on developing multidisciplinary design optimization (MDO) capabilities that automate the design process, yet not a single aircraft in service has been designed using these techniques. [88] Although there are many reasons for this lack of application, the following are commonly viewed as the most significant obstacles to the widespread application of advanced design methods to real-world problems.

1.3.1 Difficulty of establishing relevant figures of merit

In the heyday of the aerospace industry, engineers designed with the concept of "farther, faster, better" in mind - performance was the ultimate objective. In today's environment, the situation is not as well defined, and factors such as economics, environmental impact, and time to market are equally– if not more– important than achieving the best possible performance.

These additional objectives make it much more difficult to determine which aircraft is "best" overall, because most of them conflict: a faster airplane usually costs more than a slow one; an airplane with a greater range will be heavy; and so on. Figure 7, the classic illustration of this principle, shows the ideal airplane from the perspective of each engineering discipline.

Though each of the airplanes shown in Figure 7 excel in one disciplinary metric,

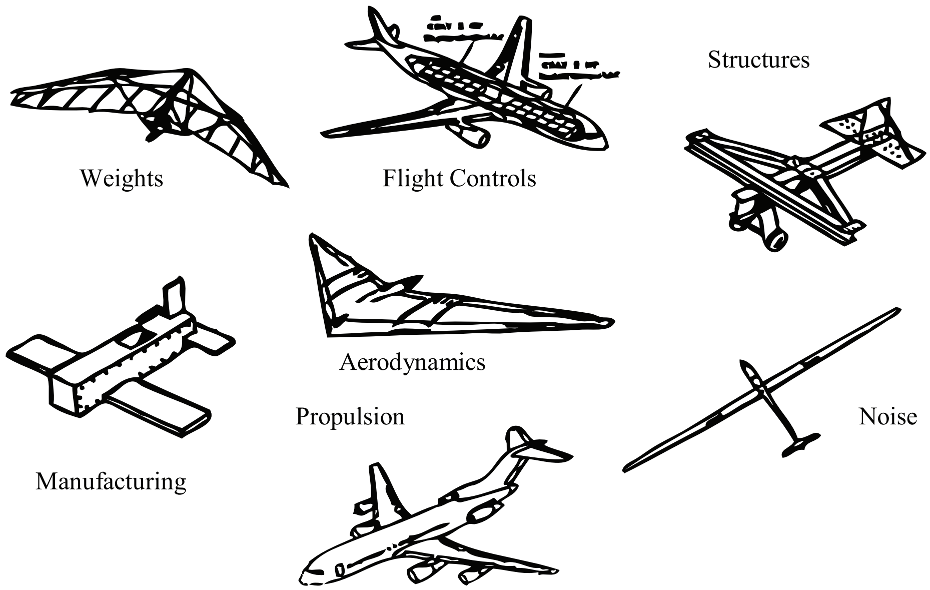


Figure 7: "Optimum" aircraft from a disciplinarian's view [88]

none are actually ideal from a systems engineering perspective because a successful vehicle will be the result of appropriate compromises between the relevant objectives. The difficulty rests in determining an appropriate compromise and formulating a single objective that encompasses the designer's preference.

For routine design problems, the inter-disciplinary relationships that form these compromises are well understood. For instance, increasing a wing's aspect ratio has a positive effect on aerodynamics owing to decreased induced drag, but a detrimental effect on wing weight. This tradeoff is well documented, and traditional aircraft design methods are typically adequate to assist the engineer in selecting an appropriate aspect ratio given the design requirements. [127] Unfortunately, this type of information is often not available for revolutionary system design problems.

Creative design problems may also introduce new metrics with which the engineer is not familiar and that traditional methods are not able to solve. In [5], Aronstein noted:

> In a traditional evolutionary design problem, experienced engineers know approximately what a near-optimum design will look like. Traditional sizing and preliminary design methods provide fine-tuning. This is not the case in a non-traditional problem where experience is lacking and where there are driving physical phenomena that are not captured by traditional aircraft design methods.

He gives an example where the imposition of a non-traditional constraint like sonic boom loudness has a large effect on other design requirements due to a coupling between the lift and volume distribution that is not present for conventional aircraft.

1.3.2 Difficulty of quantifying many important criteria

Every model used to aid in conceptual design is by definition a simplified representation of reality. Optimizers have an amazing tendency to "break" these simplified analyses: they will quickly exploit the assumptions used in the model in an effort to wring every last ounce of capability from the vehicle. This exploitation results in vehicles that appear to be optimal to the optimizer, but are impractical in reality. For example, a case is given in [88] where the optimization of a Cessna resulted in a lighter vehicle that in fact was not airworthy because the optimizer exploited the analysis' simplified stability calculations. (Figure 8)

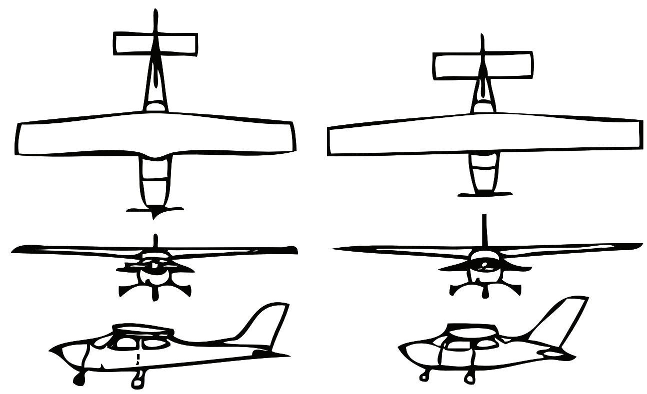


Figure 8: “Optimized” Cessna that takes advantage of a lack of stability-calculation fidelity [88]

For many problems, it may prove to be extremely difficult or impossible to create models capable of predicting every important customer requirement. Although tools for evaluating parameters such as drag and weight are readily available, metrics such as aesthetics, parts commonality, and maintainability are more difficult to analytically predict. Some researchers have been successful in creating models of these difficult-to-quantify criteria, but they are usually highly specialized and only applicable to a small subset of design alternatives. [98]

In other cases errors result from attempting to obtain results from an analysis operating outside its intended domain, or from omitting a difficult to quantify metric entirely. These errors can be mitigated by tightly restricting the design space the optimizer is allowed to search, but this eliminates many of the reasons for using optimization in the first place.

1.3.3 Computational expense of high-fidelity analysis

With enough analysis fidelity and development effort, many of the issues mentioned in the preceding section can likely be overcome. The NASA High Speed Civil Transport (HSCT) design program of the 1990s placed great emphasis on using MDO methods with tightly coupled, highly sophisticated analyses like Computational Fluid Dynamics and Finite Element Models. A large amount of research was also conducted to develop codes for calculating important but less traditional conceptual design metrics like life cycle costs and manufacturability. Together, these methods ensured realistic and high performance configurations resulted from the HSCT design process. [133] Despite this success, these techniques were never able to be used further upstream in the design process to assist in requirements and configuration selection, and after the fact analysis of the HSCT program concluded that many of the program's difficulties resulted from premature requirements and concept specification. [1]

Furthermore, direct application of these sophisticated methods to the early stages of concept formulation is impractical because of the extreme computational expense required: a single design cycle using the fully coupled HSCT4.0 environment took days to execute even on a supercomputer. [133] In addition to the computational expense, the number of man-hours required to operate such a complex simulation can be prohibitive.

1.3.4 The Curse of Dimensionality

Before the advent of modern optimization techniques, designs were traditionally "optimized" through the use of graphical methods, in which the objective (typically gross weight) was plotted as a function of two design variables, such as thrust- to- weight and wing loading. By superimposing the constraints, the analyst could find the combination of the two design variables that resulted in the lightest feasible design. However, visualization limitations prevented the analyst from exploring the impact of more than two variables at a time, resulting in designs that were not truly optimal.

MDO techniques were expected to eliminate this restriction, but have in fact only alleviated it. [28] Because the number of function evaluations required to find an optimal solution usually increases as at least a polynomial function of the number of design variables, the designer is forced to limit the number of parameters that are optimized, typically to less than twenty design variables. The designer must therefore fix potentially significant parameters at nominal values, resulting in sub-optimal performance.

1.3.5 "Noisy" and discontinuous objective functions

Most efficient optimization techniques rely upon gradient information of some form. When available, the gradient allows for rapid solution convergence, but gradient-based methods cannot be applied to the synthesis problem because the variables consist of categorical parameters such as the number of engines or type of tail to use in addition to continuous parameters such as wing sweep or area.

Even when dealing with only continuous input variables, gradient based techniques may have problems owing to noisy objective functions. Many of the models used as objectives in optimization problems require internal iteration, and this can lead to numerical noise unless convergence tolerances are very strict. In some cases, the analysis routine may fail to return an answer at all. As an example, Figure 9 displays a contour plot of sonic boom loudness as a function of thrust-to-weight and wing loading. The response is quite poorly behaved, with numerous local minima and discontinuities. Additionally, there is no data in the lower left corner of the design space because the analysis routine failed due to insufficient thrust at top of climb. Calculus-based optimization techniques are unable to cope with these problems because they are unable to escape local minima.

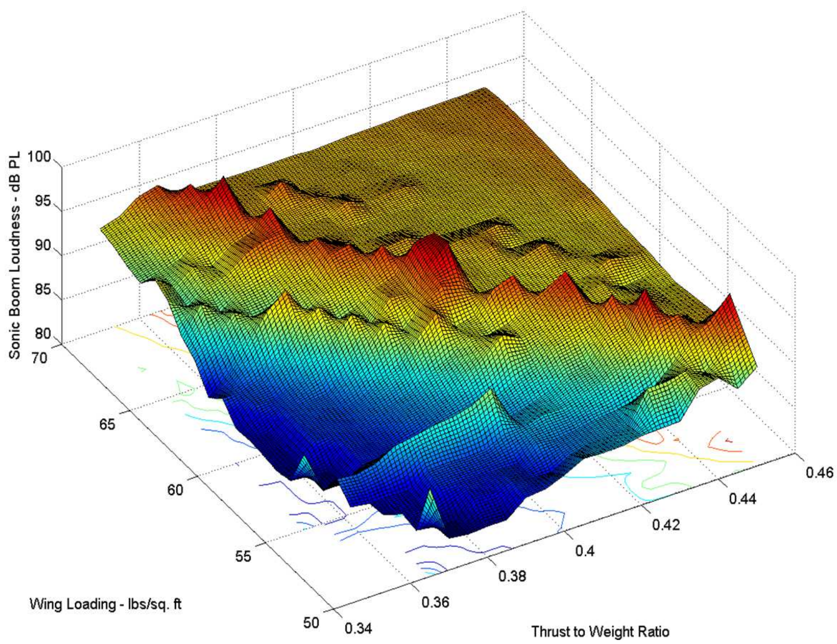


Figure 9: Example of a numerically "noisy" system level response

1.3.6 Difficulty of capturing systems-requirements-technology interactions

The role of technologies also plays a crucial role in the synthesis stage of design. According to Raymer:

> "Before a design can be started, a decision must be made as to what

> technologies will be incorporated. If a design is to be built in the near future, it must use only currently available technologies as well as existing engines and avionics. If it is being designed to be built in the more distant future, then an estimate of the technological state of the art must be made to determine what technologies will be available at that time." [127]

The incorporation of future technologies is not without risk. Because they have not yet been proven, the expected benefits of future technologies may never be realized, leading to program failure for vehicles that were designed to rely on that expected increase in performance. This risk tends to make aircraft designers consider at most a handful of unproven technologies to incorporate into a design effort, resulting in a difficult choice: which of the many technologies under development should be designed into the system?

This decision must be made during concept and requirements definition, or significant effects may be overlooked in the analysis: in the previously discussed space shuttle example, the invention of a new technology (i.e. the silica tiles) enabled the delta configuration that had to that point been infeasible due to heating constraints. A similar example from [88] shows that the addition of a stability augmentation system by itself does very little to improve an airplane's performance, but when the vehicle is re-designed to take advantage of the new technology much more significant improvements are obtained.

1.4 Dissertation Overview

This introductory chapter has provided a brief description of the aircraft synthesis process as it exists today, and discussed deficiencies with current practice. Chapter two surveys the literature and summarizes the efforts of many researchers to improve the conceptual design process. Chapter three states the hypotheses of this work, and raises several research questions to be answered by this work. The approach to achieve these goals and the methods that were developed in response to the research questions are described in chapter four. Chapter five discusses the concept exploration technique developed through the use of these methods. Finally, chapter six describes the application of the new method to the concept exploration of a supersonic business aircraft problem, and describes the modeling environment that has been created to quantify vehicle attributes.

CHAPTER II

RELATED WORK

The need to improve the traditional conceptual design process has been recognized by many researchers. Most efforts have focused on increasing the amount of information and freedom available to the designer at the earliest stages of the process. The following chapter provides a brief overview of some of the work done to date to improve the design process by members of the aerospace and other technical communities.

2.1 Requirements Analysis

The importance of requirements in the conceptual design process is universally recognized, yet many design programs have placed the task of requirements selection outside the design process by specifying them up front and freezing them. Several researchers have recognized the problems associated with this approach and have developed methods that allow the decision makers to incorporate requirements selection into the conceptual design process.

2.1.1 Unified Tradeoff Environment

Baker [8] addressed the problem of including requirements analysis in conceptual design through the development of a Unified Tradeoff Environment, or UTE. The UTE is a method for modeling system level responses as a function of the vehicle's requirements, characteristics, and technology levels. This is accomplished by creating a surrogate model for each class of variables, and then combining the three sets of equations. The results of the UTE can then be visualized in an interactive environment that allows the decision maker to vary each of these attributes and view the impact in real time. (Figure 10)

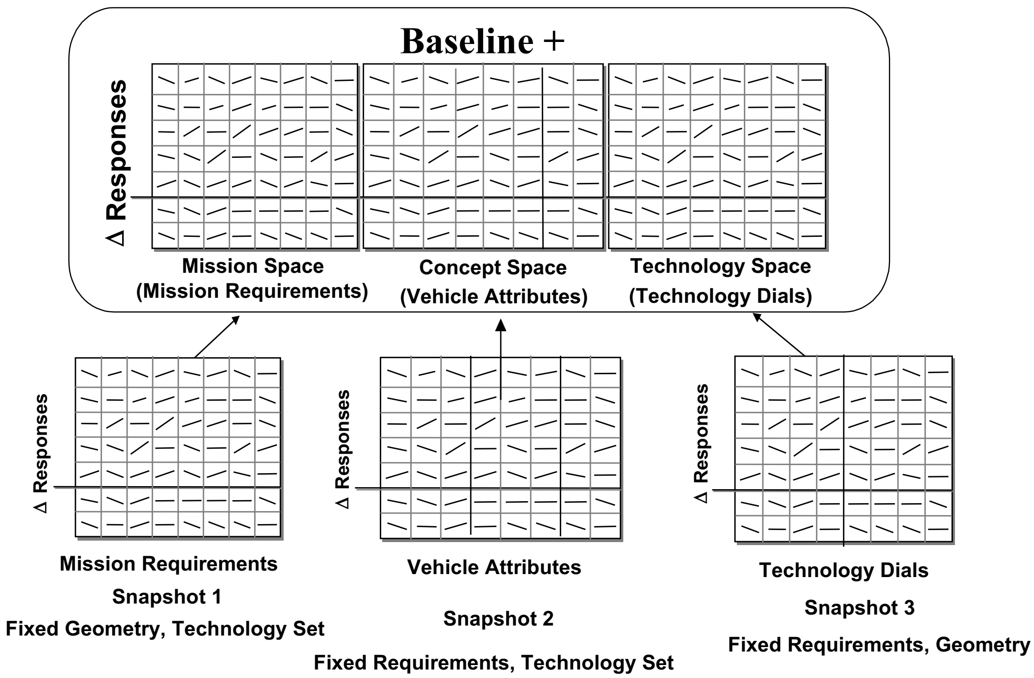


Figure 10: Example of a Unified Tradeoff Environment [8]

The Unified Tradeoff Environment was used to investigate the impact of requirements and technologies on the Future Transport Rotorcraft. [8] Although the UTE allows the decision maker to capture interactions within a category, the de-coupling of the three categories of variables prevents investigation of inter-category interactions, such as one between cruise Mach number and wing sweep. It does not solve the critical problem of addressing interactions between continuous and categorical variables such as cruise Mach - landing gear type. Because it relies upon surrogate models, it can only be used with systems that have well-behaved responses.

2.1.2 Requirements Controlled Design

Hollingsworth [69] also recognized that requirements play an extremely important role in system design and attempted to address the deficiencies of available methods through a new technique named Requirements Controlled Design (RCD). This technique is based upon catastrophe theory, and attempts to locate a system's technology boundaries, defined as the level of requirements beyond which it is impractical for one technology to function.

Requirements Controlled Design relies upon a modified Strength Pareto Evolutionary Algorithm [164] to locate these technology boundaries for each system type under consideration. Changes to the original SPEA include a modified definition of dominance and a new method of clustering. Once results have been generated, a Gaussian process metamodel of the system-level responses is created to aid in visualization of the requirements hyperspace. The method was applied to the system identification of a lightweight helicopter design problem, and revealed several interesting relationships between the vehicle's configuration and requirements. However, several issues were encountered during the research. Hollingsworth noted that the study's reliance on a highly simplified analytical model diminished the usefulness of the results, and also that the need to re-run the MSPEA algorithm for every system alternative limited the applicability of the method to problems with only a handful of alternatives.

2.2 Subjective Synthesis Methods

Analytical methods have typically seen little application to synthesis problems. Instead, engineers have most frequently used intuition, past experience, or relied upon historical precedence within their organization to make decisions during design synthesis. It is therefore not surprising that the earliest attempts to improve the conceptual design process focused on providing a more structured framework that would serve to organize the available information and thereby foster creativity.

2.2.1 Morphological Analysis

The method of Morphological Analysis was developed by Fritz Zwicky in the late 1940's to study the relationships between components of large and difficult- to- quantify problems through decomposition. [169] It consists of five iterative steps, listed in Table 1. The method is designed to encourage creativity by ensuring that no concept is "discarded a priori as being unimportant." After all, according to Zwicky, "within the final and true world image everything is related to everything," and therefore all relevant concepts are worthy of consideration.

Table 1: The process of Morphological Analysis

Step	Task
1	Formulate the problem.
2	Establish relevant parameters and alternatives
3	Construct a multidimensional Matrix of Alternatives, containing all possible solutions
4	Evaluate all consistent solutions
5	Select the optimal solution

Morphological Analysis has been applied to a truly large variety of problems, from policy analysis [130] to the development of propulsion systems. [168] It successfully promotes creativity and innovation by efficiently displaying massive amounts of potential solutions and encouraging the designer to think in terms of "why not" rather than "why." However, its ability to display myriad solutions can be problematic. By offering such an abundance of alternatives, this type of analysis can easily overwhelm the designer. For example, Table 2 presents a Matrix of Alternatives containing more than 120,000 possible configurations for a supersonic business jet. Because time and resource limitations prevent the designers from evaluating more than a handful of these alternatives, they may be tempted to make hasty decisions that later prove to be incorrect.

One final problem with Morphological Analysis is its limited capacity to address the hierarchical problems that are frequently encountered in engineering design. For example, each of the wing configuration options listed in Table 2 has a number of descriptive parameters associated with it (ie. sweep, taper ratio, etc.) These descriptive parameters are typically not included in the Morphological Analysis because they would cause the size of the Matrix of Alternatives to balloon out of control. However, the omission of these parameters can lead to sub-optimal conclusions being drawn from the analysis. (Figure 11)

2.2.2 Integrated Product and Process Development

Integrated Product and Process Development, or IPPD, is one of the most widely used methods developed to date for improving the design synthesis process. Inspired by the quality methods pioneered by Taguchi and others in Japan in the late 1960's,

Table 2: Matrix of Alternatives for a small supersonic transport containing 120,960 possible configurations

Planform Type	Delta	Double Delta	Ogee	Blended	Swing-wing	Strut-braced	Unswept NLF
Wing Location	Low	Mid	High				
Pitch Control	Horizontal Tail	T-Tail	Canard	Three-surface	Tailless		
Engine Cycle	Existing moderate BPR turbofan	New mixed flow turbofan	Variable cycle				
Powerplant Installation	Under wing	Fuselage mounted	Tail mounted	Tri-jet			
Inlet Type	Axisymmetric	2-Dimensional					
Inlet Compression	Mixed	External					
Inlet Geometry	Fixed	Variable					
Nozzle Type	Convergent-Divergent	Mixer-Ejector					
Materials	Aluminum	Composite					
High Lift System	None	Plain Flaps	Fowler Flaps				

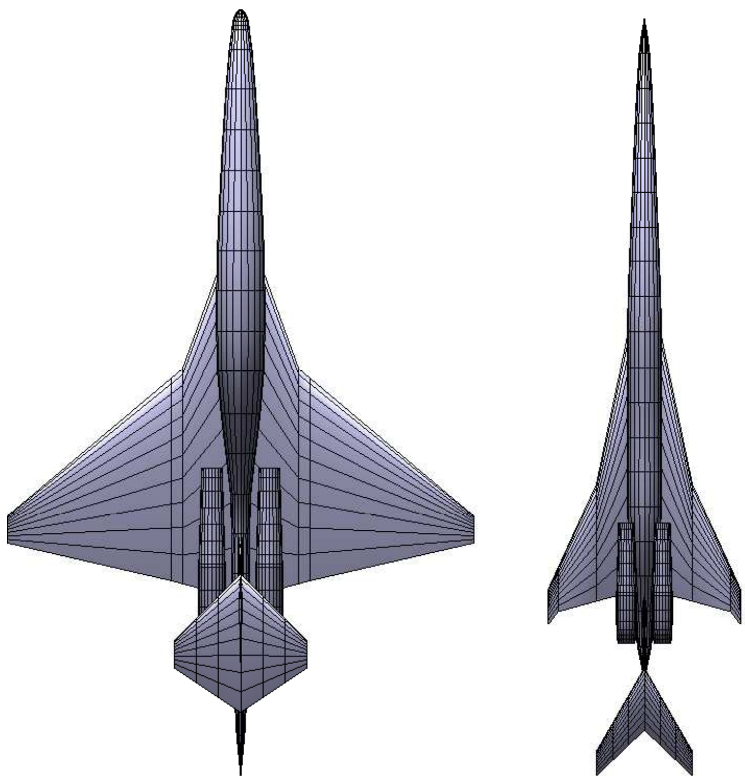


Figure 11: Two configurations with the same morphology but very different attributes

IPPD improves upon the traditional, serial design process by using Integrated Product Teams (IPTs) to consider all aspects of the design at the earliest stages of development. [22] As depicted in Figure 12, this results in more knowledge and less cost commitment early on.

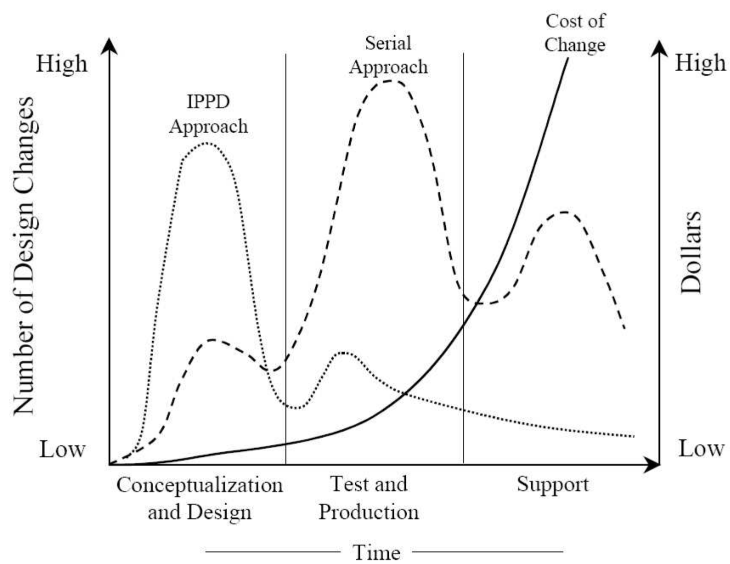


Figure 12: Comparison of the traditional and IPPD design processes [101]

Though the steps of IPPD are rather similar to that of the conventional design process, the tools used to aid in decision making are rather different. These tools include brainstorming techniques such as affinity diagrams [43] and schematic block diagrams [42] that help to perform a functional analysis and decomposition of the system. Another important tool used during the IPPD process is Quality Function Deployment (QFD), which makes use of the subjective input of the analysts and a “House of Quality” (Figure 13) to focus the attention of the design team on critical engineering characteristics. In the QFD process, the customer typically provides a list of requirements and relative importance values for each metric. Cause and effect diagrams are used to determine a list of appropriate engineering characteristics and their relationships. These characteristics are then placed into the "roof" of the House of Quality. Each box in the main "room" of the House is then filled in with a symbol that represents the degree of relationship between the customer requirements and engineering characteristics. This information is used in conjunction with a list of subjective difficulty ratings to provide an estimate of the relative importance of each engineering characteristic on fulfilling customer requirements.

The results of Quality Function Deployment are typically used in combination with other techniques such as Pugh Decision Matrices [35] to qualitatively compare different design alternatives and rank them according to how well they fulfill the requirements. Some versions of the IPPD methodology also incorporate modeling and simulation tools to evaluate the system alternatives that emerge from concept downselection.

IPPD methods have been successfully applied to many design problems ranging from CD jewel cases [43] to software programs [92]. The method has also proven useful at broadening the decision-makers understanding of the relationship between engineering characteristics and customer requirements. One weakness of the technique, however, is the lack of an effective mechanism to perform concept downselection when there are a very large number of feasible alternatives.

2.2.3 The Theory of Inventive Problem Solving

The theory of inventive problem solving, or TRIZ [2], was developed in Russia in the late 1940's by Genrich Altschuller, who studied the patent library in order to

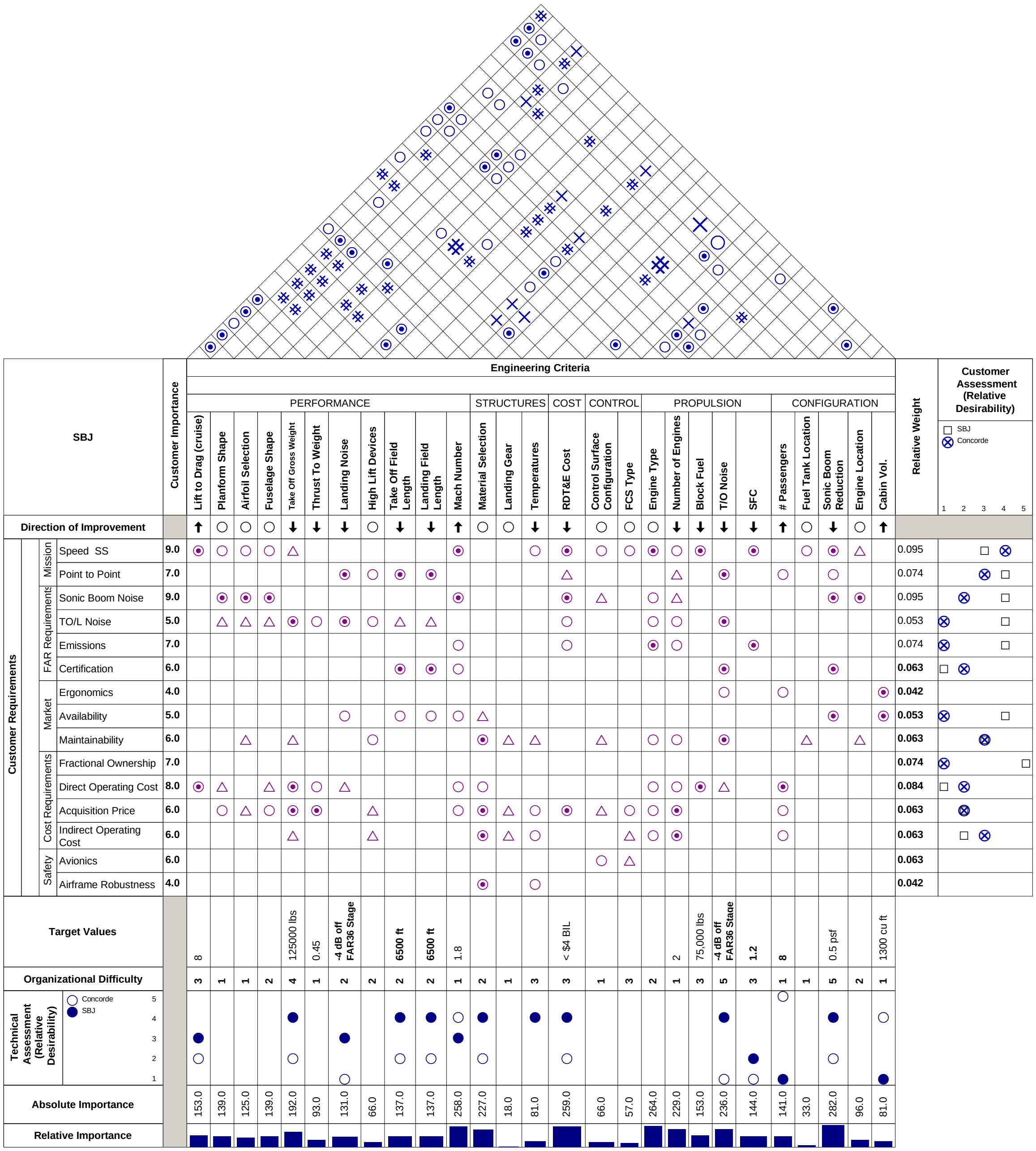


Figure 13: "House of Quality" for a supersonic business jet design [20]

determine the manner in which inventors and engineers tackled technical problems. During his research, Altschuller discovered that problem solutions could typically be classified into five groups: routine design, minor variations, fundamental improvements to resolve contradictions, solutions resulting from new scientific discovery, and pioneering inventions. The TRIZ method was developed to assist engineers in resolving the third type of problem by compiling a comprehensive list of the inventive principles used to address the common objectives. Application of TRIZ has resulted in many improved products [2], but the method is difficult to use when no existing product is available to improve upon, as is the case during revolutionary design.

2.2.4 Expert Systems and Case-based Design

Beginning in the 1980s, a number of researchers investigated whether artificial intelligence (AI) techniques could be used to assist in decision-making during the conceptual design process. A large number of different methods including knowledge-based expert systems and case-based design were developed as a result of this research.

One of the most popular artificial intelligence design methods is knowledge-based expert systems, defined in [83] as "computer programs that combine detailed domain facts (the information pertinent to a particular application subject) with heuristic rules (essentially rules of thumb) to enable problem-solving performance in some technical field that is at least equivalent to the performance of an expert."

A number of frameworks for expert systems have been developed, including CLIPS [79] and Engenious [155]. Expert systems have been most frequently applied to classification problems and as a tool to assist and troubleshoot software, an example of which is the infamous Microsoft Office paperclip assistant. They have also been applied to several aircraft design problems with varying degress of success.

The author of [155] successfully used expert systems in conjunction with numerical optimization techniques to develop turbine blades with improved efficiency. It was found that using an expert system with numerical optimization produced "conventional" designs, but coupling numerical optimization with genetic search produced designs with greater performance that had "parameter distributions opposite to what is done traditionally." In [89], an aircraft optimization system was developed that used expert systems to warn the user when the code was being operated outside of its bounds. The system was also used to suggest remedies when performance constraints were not met; an example of such a suggestion would be "Take off field length constraint not met, consider increasing thrust or wing area."

Case-based design is a rather different approach to artificial intelligence, because rather than encapsulating knowledge in the form of "if-then" statements, it seeks to solve problems by searching a database for previously solved problems with similar descriptions and then adapting them to meet the new requirements. Instead of explicitly encapsulating knowledge (i.e. saying that a turboprop would be a poor choice for a Mach 2.0 fighter) the case-based design system would implicity come to this conclusion because its database would not contain any previous Mach 2 designs with such a powerplant. In [129], a case-based design system called AIDA was developed to assist in conceptual aircraft design. The method demonstrated using the design of a 70-passenger regional jet transport, and resulted in a configuration that was a hybrid of the Fokker 70 and the BAe RJ70. The work concluded, however, that "Design which comprises new components and new parameters, classified as creative design, does not seem to be suitable for support by case-based reasoning".

In recent years, there has been diminished interest in expert systems from the aerospace community. This is likely because "tremendous amounts of knowledge are needed to provide a sufficient basis for intelligent behavior" [83] and "transferring human knowledge into expert systems syntax is often a very difficult task" [155]. It seems that the most promising area for the future use of expert systems in aircraft design may be to give advice regarding which optimization and analysis strategies are appropriate for a given problem.

2.3 Quantitative Synthesis and Optimization Methods

Though each of the subjective synthesis methods described in the previous section have been successfully used to improve the conceptual design process by providing a framework to aid in decision making, none can easily solve problems for which the designer wishes to incorporate simulation into the concept selection process. Because of this, subjective methods may suggest sub-optimal designs if the design team's assumptions about achievable performance levels are incorrect. The desire to incorporate simulation and decrease design cycle time has led to the development of a large number of optimization techniques applicable to different classes of design problems. In all cases, however, the goal of optimization is either to maximize or minimize some function $f(\bar{x})$ by varying $\bar{x}$ subject to constraints $g_j(\bar{x})$.

2.3.1 Local optimization methods

Local optimization methods can be applied when a function is known to have only one local optimal value. Despite this limitation, they are the most commonly used of all optimization techniques, and a large number of different algorithms to optimize functions with one optimum have been developed. A sample of these methods is given in Table 3. The local optimization method most appropriate for a given problem depends upon a number of factors, including whether or not the function is continuous, differentiable, or constrained. Gradient-based methods such as Sequential Quadratic Programming and Fletcher-Reeves Conjugate Gradient method are typically the most efficient if the function is continuous and smooth, while methods such as the Nelder-Meed Simplex method or Powell's Conjugate Direction method may be more efficient for problems which are less well-behaved. The benefits and drawbacks of each of these algorithms has been covered in depth in the literature. [53]

Table 3: Popular local optimization methods

1	Fletcher-Reeves Conjugate Gradient Method [54]
2	Nelder-Meed Simplex Method [114]
3	Method of Feasible Directions [167]
4	Powell's Conjugate Direction Method [124]
5	Sequential Linear Programming [93]
6	Sequential Quadratic Programming [93]
7	Tabu Search [59]

2.3.2 Multidisciplinary Design Optimization methods

In [58], Multidisciplinary Design Optimization (MDO) is defined as *A methodology for the design of complex engineering systems and subsystems that coherently exploits* *the synergism of mutually interacting phenomena.* Although nearly all design problems are multidisciplinary, until recently individual experts or groups of experts often performed disciplinary analysis in a serial fashion, preventing full exploitation of synergistic interactions.

Multidisciplinary Design Optimization is not an optimization technique in itself, but rather a method for analysis integration and decomposition. MDO techniques facilitate the evaluation and optimization of complex systems of analyses. These systems may be represented by N^2 diagrams, an example of which is presented in Figure 14. The N^2 diagram is a convenient representation of the necessary information flow between disciplinary Contributing Analyses (CAs). In many cases feedback in an N^2 diagram can be minimized via rescheduling, but oftentimes it is impossible to avoid feedback completely. A variety of methods have been proposed for facilitating optimization of these coupled systems, including Collaborative Optimization [19], Bi-Level Integrated System Synthesis [142], Optimizer-Based Decomposition [141], and Fixed Point Iteration. Two of the most commonly applied MDO methods are discussed below.

2.3.2.1 Fixed Point Iteration

Fixed point iteration (FPI) is the oldest method for optimizing a coupled system of analyses. When using FPI, an initial guess is made for each feedback required in the N^2 diagram, and iteration is used to converge each of these guesses to be consistent within a specified tolerance. Although this method of analysis is simple, it has several drawbacks. If the analyses are computationally expensive, the use of

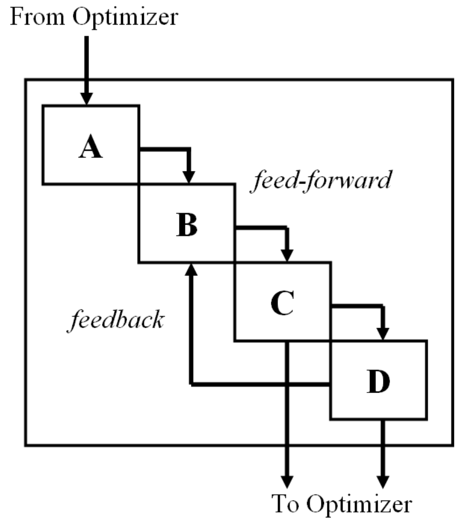
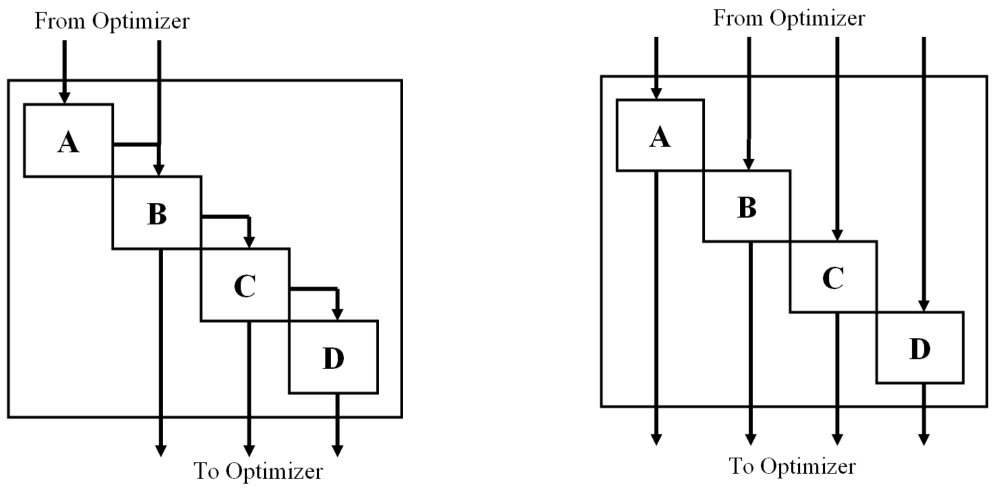


Figure 14: N^2 Diagram for an example MDO problem

FPI can lead to very large computational cost. This problem is exacerbated if FPI is used in conjunction with a gradient-based optimizer because without extremely tight convergence tolerances the calculated gradients may be very inaccurate.

2.3.2.2 Optimizer-based decomposition

Optimizer-based decomposition is a more recent method that focuses on reducing iteration by breaking the feedbacks and feed-forwards in the N^2. Under OBD, new intermediate variables are introduced that provide values to the contributing analyses, and compatibility constraints are used to ensure that these intermediate variables are consistent at the end of the optimization process. This procedure is illustrated in Figure 15. The execution time of each system-level function call will be greatly reduced because internal iteration is no longer required, but the addition of the compatibility constraints increases the workload of the optimizer and will require more iterations. Breaking feed-forward variables is only advantageous if multiple computers will be used to evaluate the different contributing analyses, and therefore partial OBD is often used to break only the feedback links.



(a) (b)

Figure 15: N^2 diagram for a (a) partial and (b) full Optimizer-Based Decomposition implementation

2.3.3 Technology Integration, Evaluation, and Selection

Technology Identification, Evaluation, and Selection (TIES) is a method by which a designer can investigate the impact of future technologies on system feasibility and viability. [85] Given a reference design, the method uses approximation and optimization techniques to determine if project goals can be met with present day-technologies. (Figure 16) If this proves to be impossible, the method can be used to perform a top-down technology assessment via a Technology Impact Forecast, or TIF. The TIF is a dynamic environment based upon a surrogate model of physics-based analysis tools that lets decision-makers assess the relative impact of advanced technology factors, or K-factors, on system-level metrics.

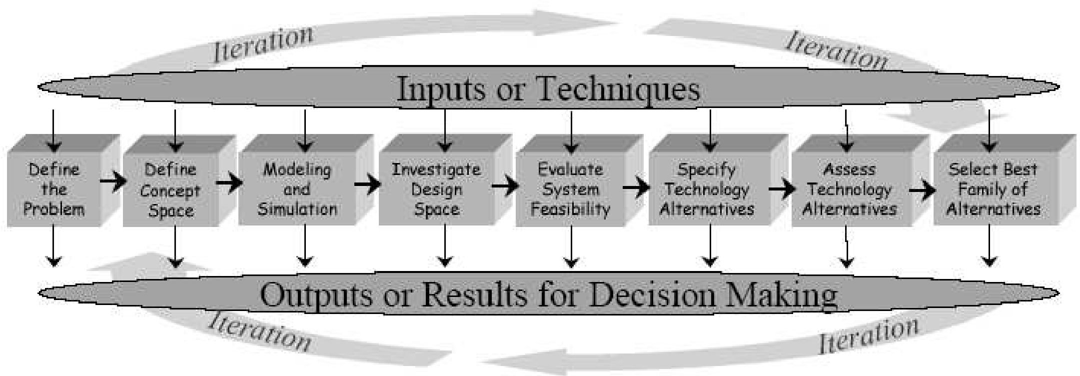


Figure 16: TIES Methodology Flowchart [85]

If a portfolio of technologies under development is known, TIES can be used to perform a bottom-up analysis, thereby finding a "best set" of technologies to invest in. Portfolio selection requires detailed knowledge of the expected benefits and degradations associated with each technology, but because the method relies upon approximate models it is very simple to incorporate uncertainty analysis via Monte Carlo Simulation. This fact is very important because of the uncertain nature of technologies which have not been fully developed.

The TIES method has been successfully applied to a diverse set of problems including supersonic transport and propulsion system design. One deficiency of the method is that it does not facilitate exploration of the concept matrix, and may therefore lead to designs that require excessive technology investment in the case where the baseline design was a poor one. Another issue is that the TIES method does not account for the interactions between the technology portfolio and the selected platform. This type of interaction, such as the one between wing planform type and thermal protection system on the space shuttle, can have very significant implications.

2.3.4 Genetic Algorithms

The Genetic Algorithm is a type of Evolutionary Algorithm that mimics Darwin's evolutionary process and uses nature-inspired operators to evolve designs of improved performance. Originally developed by John Holland in the 1960s [68], the algorithm typically operates on a population of individuals, and over several generations the principal of "survival of the fittest" is employed to bring about favorable change among population members.

2.3.4.1 The classical Genetic Algorithm

The classical GA described in [60] has five basic steps: initialization, evaluation, selection, crossover, and mutation. (Figure 17) The algorithm begins with the creation of an initial population of designs, which are usually chosen at random. These designs are encoded as binary strings that are concatenations of the binary conversion of each decision variable. Once encoded, each of the designs is then evaluated and assigned a "fitness" value, analogous to the objective value used in more traditional optimization approaches. This is typically the most computationally expensive portion of the algorithm for most engineering applications because each design needs to be assessed using the simulation environment.

After evaluation, the population undergoes a process known as "selection," during which parents for the next generation are chosen, with a preference for designs of better "fitness." There are multiple ways to accomplish this, but the most common methods are known as tournament selection and roulette wheel selection. Under tournament selection, a series of competitions are held in which n solutions are chosen

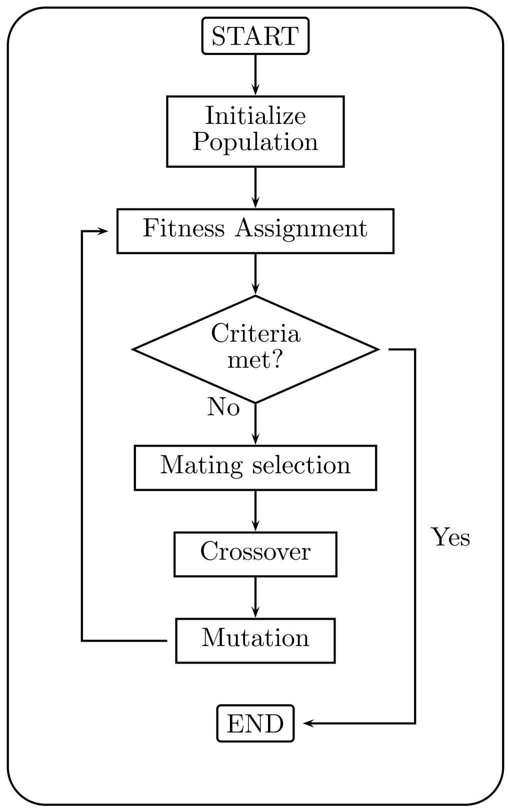
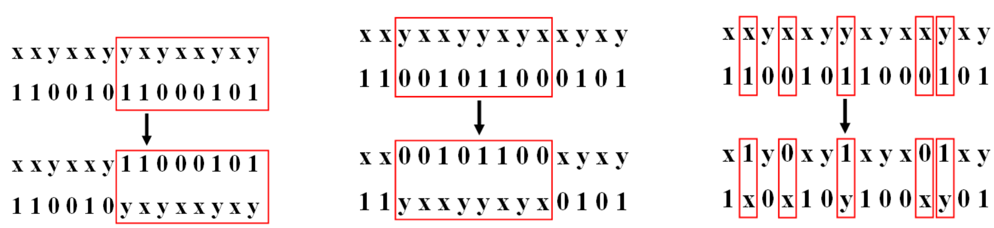


Figure 17: The simple Genetic Algorithm

from the current population, and only the individual with the best fitness is chosen as a parent for the next generation. The result of this operation is that designs of better fitness occur more frequently in the mating pool. When roulette wheel selection is used, the sum of the fitness of all population members is calculated, and each individual is assigned a proportion of the mating pool equal to the ratio of its fitness to the total.

Once the mating pool has been established, the genetic material of the parents is combined to form children designs during recombination. In the simple Genetic Algorithm, this recombination occurs via a single point crossover operator that acts by randomly selecting a splice point in the binary string and then swapping bits between the parents at the splice. (Figure 18(a)). Several other alternatives to the 1-point crossover operator exist, including the 2-point and uniform crossover operators. (Figure 18) The benefits and drawbacks of each of these recombination operators are thoroughly discussed in literature. [143] [159]



(a) 1-point (b) 2-point (c) uniform

Figure 18: Genetic Algorithm crossover operators

Following the recombination step, an operator known as mutation is typically applied that infrequently changes the value of a bit with a specified mutation probability. The mutation operator helps to preserve diversity in the population and increase the likelihood that the true global optimum is found by the algorithm. This entire process, known as a *generation*, is repeated until a user-specified criteria such as a maximum number of generations or performance threshold is reached.

2.3.4.2 Theory of Genetic Algorithms

The foundation of evolutionary algorithm theory is the schema theorem, first proposed by Holland. [68] A *schema*, or building block, is defined as a set of chromosomes that share certain values. The *order* of a schema is the number of defining values that compose it, and a schema's *length* is the distance between its outermost values in the bit string. As an example, consider a 6-bit binary string. A possible schema of this string would be *1*00*, representing all six bit strings containing a 1 in the second position and 0 in the fourth and fifth positions. This schema is of order 3, and has a length of 4.

The schema theorem states that "Short, low-order, above-average fitness schemata receive exponentially increasing trials in subsequent generations." Assuming 1-point crossover and roulette wheel selection, this can be mathematically expressed as:

$$m(h, t+1) \geq \frac{m(h,t)f(h)}{\bar{f}_t}[1 - P_c\frac{\delta(h)}{l-1} - O(h)P_m] \tag{1}$$

where $m(h,t)$ is the expected number of schema h at generation t, $f(h)$ is the fitness of schema h, $\bar{f}_t$ is the average fitness at generation t, l is the genotype length, $\delta(h)$ is the defining length and $O(h)$ the order of schema h.

For problems with no interactions, each schema will be of length one. However, if the fitness contribution of one bit is dependent on that of one or several others, this will increase the length of the schema and delay its rate of expansion within the population. Examination of Equation 1 also reveals that it is advantageous to place parameters which may have interaction terms closer together in the bit string, as this reduces schema length $\delta(h)$.

Holland's building block hypothesis is closely related to the schema theorem, and states that "short, low-order, highly fit schemas recombine to form even more highly fit, higher-order schemas." Because a single schema does not completely define a problem, the fitness of a particular building block is dependant on that of the others used to form the bit string. This results in each building block having a distribution of fitness values within a given population population, the mean of which may vary depending upon the sample, as depicted in Figure 19. In problems with large bit strings and many building blocks, the variance on the mean fitness value of each building block will be rather large, such that there is a significant chance of mis-ordering and in fact selecting an inferior schema.

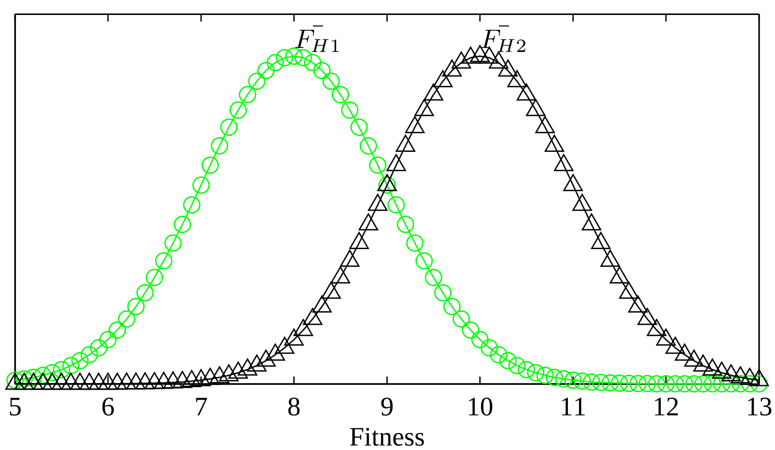


Figure 19: Distribution of mean fitness values for individuals containing two distinct building blocks

This chance of mis-ordering is reduced as population (or sample) size is increased, and the Gambler's Ruin problem has been used as a model to derive theoretical population sizing guidelines for simple test cases such as the OneMax problem. [24] Unfortunately, these population sizing guidelines are difficult to use for real world problems where schema order and length are unknown.

2.3.5 Advanced Genetic Algorithms

Genetic Algorithms have been widely applied to optimization problems because of their ability to handle functions with both continuous and discrete variables, their adaptability for use in multi-criteria optimization, and the fact that they do not need gradient information. They are also very effective for global optimization problems that have numerous local minima. Despite these advantages, several deficiencies with the classical GA have been identified and a large amount of research has been performed to remedy them.

2.3.5.1 Real Parameter Genetic Algorithms

One problem with the classical Genetic Algorithm is the requirement that the design be encoded as a binary string. This encoding process requires the discretization of continuous decision variables, which may prevent the algorithm from finding the problem's true optimum. This issue can be addressed by using more bits of precision, but the large string lengths associated with high-precision encodings require large population sizes and greater computational expense. [60]

These observations led to the development of the Real Parameter Genetic Algorithms that do not rely on binary encoding but instead operate on the actual decision variables. Although the schema theorem only applies to algorithms with binary alphabets, a new theory based upon interval schemata has been proposed that extends Holland's theory to real-coded GAs. [48] The advent of this new class of GA required the development of new crossover operators because the bit swap mechanism associated with the conventional GA is ineffective when applied to non-binary encodings. Several methods for real-coded recombination have been proposed, but the two most commonly used by modern GA practitioners are the BLX-α and SBX operators. [48][37]

BLX-α: The BLX-α operator was one of the earliest crossover operators developed for real-valued Genetic Algorithms. [48] This operator creates two children solutions from two parent solutions by using the relation:

$$x_i^{(1:2,t+1)} = (1-\gamma_i)x_i^{(1,t)} + \gamma_i x_i^{(2,t)} \tag{2}$$

where $\gamma = (1+2\alpha)u_i - \alpha, u_i = rand[0\ 1]$, and α is a positive number. The value of α can be used adjusted to control the nature of the search: small values emphasize exploitation, and larger values promote exploration. Several reports indicate that an α value of 0.3-0.5 may yield the best performance for many problems.

SBX: One other commonly used crossover operator in Real Parameter Genetic Algorithms is Simulated Binary Crossover (SBX). [40] This operator has been explicitly designed to function in a manner similar to the 1-point crossover operator associated with the standard genetic algorithm. The SBX operator was also designed to have two additional properties [40]:

1. The extent of children solutions is in proportion to the parent solutions.

2. Near parent solutions are monotonically more likely to be chosen as children solutions than solutions distant from parents.

The SBX operator creates two children solutions $x_i^{(1,t+1)}$ and $x_i^{(2,t+1)}$ from two parent solutions $x_i^{(1,t)}$ and $x_i^{(2,t)}$ by using Equations 3 and 4:

$$x_i^{(1,t+1)} = 0.5[(1+\beta_q)x_i^{(1,t)} + (1-\beta_q)x_i^{(2,t)}] \tag{3}$$

$$x_i^{(2,t+1)} = 0.5[(1-\beta_q)x_i^{(1,t)} + (1+\beta_q)x_i^{(2,t)}] \tag{4}$$

where β_q is drawn from the probability distribution given in Equation 5:

$$P(\beta) = \begin{cases} (2u)^{\frac{1}{\eta+1}} & \text{if } u <= 0.5, \\ (\frac{1}{2(1-u)})^{\frac{1}{\eta+1}} & \text{otherwise.} \end{cases} \tag{5}$$

and η is a parameter that controls how close the child solutions are to their parents. The result of the SBX operator is that when two similar parents undergo crossover the resulting solutions will likely be similar, while if the parents are quite different the children solutions will also be diverse. This is referred to as self-adaptation, and is regarded as a desirable crossover property.

2.3.5.2 Parallel Genetic Algorithms

One of the most frequent criticisms of Genetic Algorithms is that they typically require a very large number of function evaluations. This was not much of a problem for previous applications of GA to aircraft conceptual design because computationally inexpensive statistical methods were usually used for function evaluation. Unfortunately, revolutionary vehicle design does not readily lend itself to the use of very simplified analysis because the vehicle being studied lies outside the bounds of the historical database, necessitating the use of physics-based analysis tools for performance prediction. One commonly used approach to accelerate GA convergence is to harness the capabilities of parallel computing by using Parallel Genetic Algorithms (PGAs). These algorithms come in three main varieties: master/slave, coarse-grained, and fine-grained.

Master/Slave: The master/slave paradigm is the most intuitive of the evolutionary algorithm parallelization methods. This parallelization technique relies on a single master computer to control the genetic operators such as reproduction and selection, and multiple slave processors are used to evaluate the different population members' fitness values in parallel as shown in Figure 20.

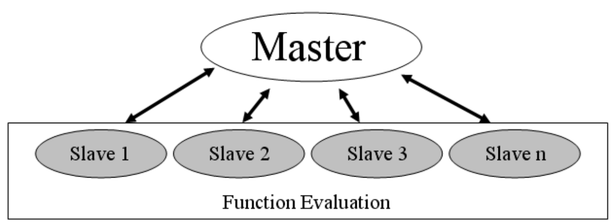


Figure 20: The Master-Slave Parallel Genetic Algorithm

Apart from the fact that fitness evaluation is performed in parallel, the Master/Slave parallel GA is identical to standard GAs. The Master/Slave paradigm is not appropriate for problems with extremely simple objective functions because overhead associated with network communication may be equal to or greater then the cost associated with the actual function evaluation. However, this method of parallelization can lead to a nearly linear speedup in the case where the fitness evaluations are time consuming.

Coarse-grained: Coarse-grained parallel GA's independently evolve multiple sub-populations but infrequently "migrate" individuals between populations. (Figure 21) This parallelization method is also known as the "island model" because it was inspired by the natural evolutionary processes that occur on island chains.

In the coarse-grained paradigm, each population is separately initialized and evolves in parallel with the other independent islands. Each of the islands may have identical or different GA parameter settings such as population size and crossover probability. The effect of this separation is that the probability the algorithm will get stuck in a local optimum due to premature convergence is decreased. Another advantage of the coarse-grained model is that it requires much less communications overhead compared to the master/slave paradigm, and may therefore be more appropriate for problems with inexpensive function evaluations.

Communication between populations can be carried out in a number of different fashions. The populations are typically arranged in some geometric shape such as a torus, ring, or hypercube. One of the primary problems with the coarse grained model is that the user is required to specify a large number of additional parameters such as:

1. The number, size, and arrangement of subpopulations

2. The magnitude and frequency of migration

3. Migrant selection and replacement policy

4. Individual island GA parameters

The selection of these parameters can have a significant impact on the results of the problem, yet they are problem dependent and no good guidelines exist. [148] Research has shown that coarse-grained PGAs can exhibit a super-linear speedup with the number of populations, but this claim is viewed as controversial within the GA community.

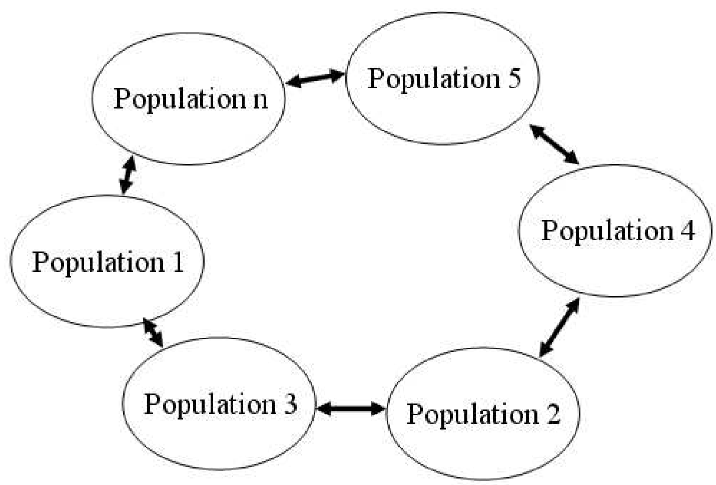


Figure 21: Coarse-grained Parallel Genetic Algorithm with a ring topology

Fine-grained: Fine-grained PGAs are designed to have superior performance on massively parallel computers which may have hundreds of processors. These algorithms organize the population into a spatial structure such as a grid similar to the one shown in Figure 22, and only allow population members to compete and mate with members located in adjacent cells of the structure. Because of this spatial partitioning, evolutionary operators such as crossover and selection can be parallelized in addition to function evaluation. Fine-grained PGAs may also exhibit more diversity than a traditional GA because of the fact that genetic operators only operate on population members within a neighborhood means that it will take more generations for an excellent individual to "take over" the population. On supercomputers with shared memory systems this method of parallelization can be very efficient, but it has the opposite effect on clustered systems that use a network for communication. [116]

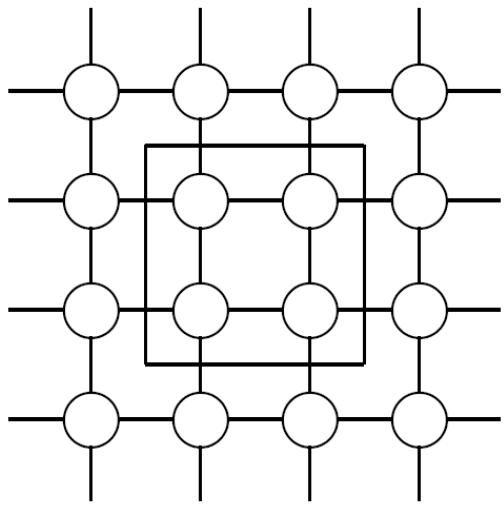
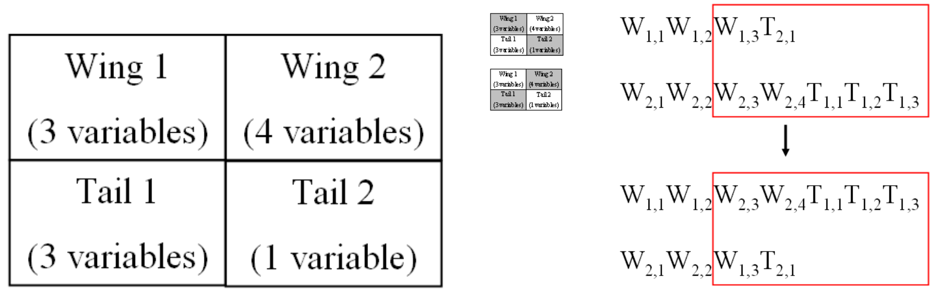


Figure 22: Fine-grained Parallel Genetic Algorithm topology

2.3.5.3 Structured Genetic Algorithms

Initial studies revealed that conventional GA crossover operators exhibit poor performance when applied to problems with hierarchical variable structures, such as those commonly found in systems engineering problems. This is because with a conventional encoding, the alleles of these hierarchical genes are not aligned. (Figure 23)

One method to address this difficulty is the Structured Genetic Algorithm (sGA). [33] Originally developed to help maintain genetic diversity within a population by



(a) Simple design heirarchy

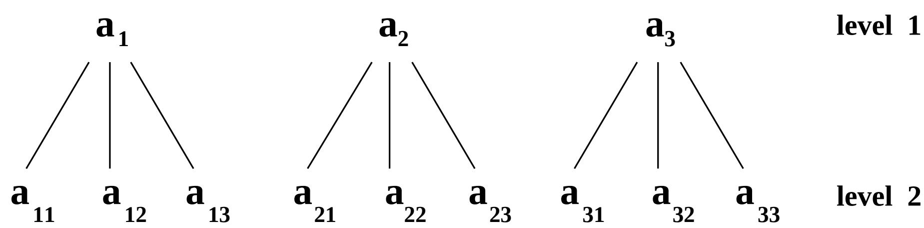
(b) Application of 1-pt crossover to a hierarchical problem

Figure 23: Unsuitability of standard genetic encoding and operators for hierarchical problems

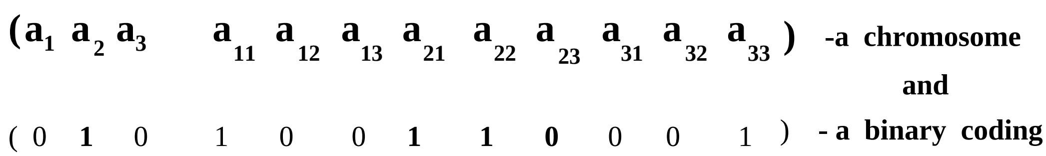
allowing it to adapt to shifting requirements, the sGA modifies the original GA's encoding scheme by introducing control genes that activate or deactivate subsequent dependent parts of the genome. The deactivated sections of the genome are not deleted, but maintained for possible re-activation in future generations. (Figure 24)

In [119], it was recognized that the Structured Genetic Algorithm could be used to search whole system design hierarchies, and the sGA was applied to the system design of a hydropower plant. The algorithm was used to optimize parameters including dam sites, material types, and modes of operation, with a total of twenty different discrete options. These discrete options also had dependent continuous variables such as tunnel lengths or period of operation. (Figure 25)

The results of this research revealed that while the sGA could efficiently locate high performance systems, the vast majority of the search was confined to a small portion of possible system alternatives. In fact, one of the twenty system architectures was never evaluated by the sGA, and several others only received a handful of function calls. (Figure 26(a)) A decision maker may therefore have little confidence in the



(a) A 2-level structure of sGA.



(b) An encoding process of sGA.

Figure 24: A Simple Structured Genetic Algorithm Encoding [33]

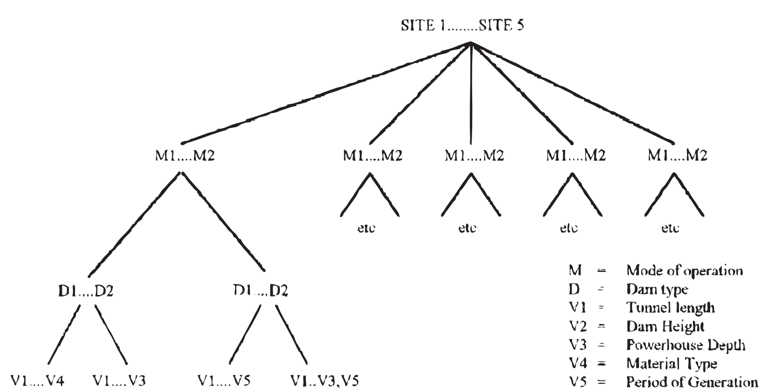
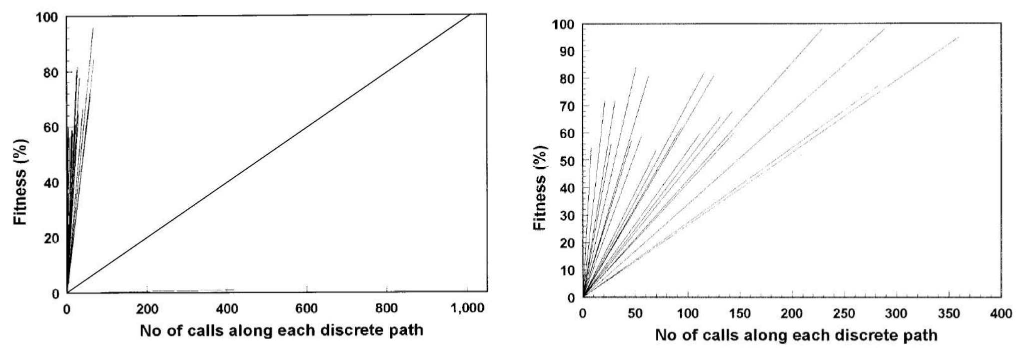


Figure 25: Structure of Parmee's Hydropower Plant Design Problem [119]

results. This shortcoming was recognized and addressed via the introduction of very high (20%) mutation rates for the discrete variables. The results using the sGA with high mutation rates yielded a much more even distribution of function evaluations across the design space. (Figure 26(b)) However, this approach may not be ideal because high mutation rates are expected to be disruptive. [7]



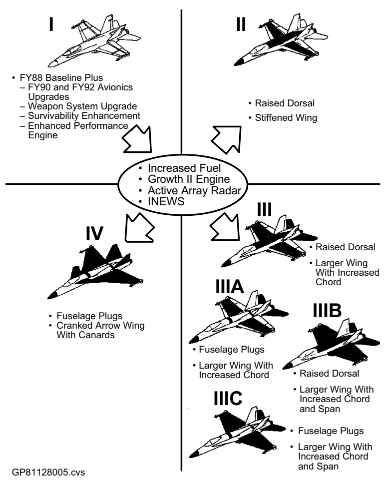
(a) sGA results after 50 generations with 2% mutation

(b) sGA results after 50 generations with 20% mutation

Figure 26: Distribution of function calls to the hydropower plant design problem using the Structured Genetic Algorithm [119]

2.3.5.4 Interactive Genetic Algorithms

Ensuring that realistic configurations are the product of the conceptual design process is one of the most prominent obstacles to the widespread application of multidisciplinary design methods to conceptual design problems. This difficulty stems from the fact that it may be extremely difficult or computationally infeasible to explicitly include every objective and constraint in the computational model. For example, Figure 27 displays system alternatives and an alternatives evaluation matrix used for the design of the F/A-18 E/F. Several of the critical criteria listed in the figure, including "carrier suitability" and "growth potential" have some quantitative aspect, but also include subjective factors. In fact, while the Super Hornet is one of the most modern airplanes in U.S. military service, MDO was not used for its design because "an objective function that could be used to determine the optimum configuration would prove very difficult to formulate in this case. In fact, typical parameters that have been suggested as objective functions such as minimum weight or minimum cost were not the final discriminators of the selected configuration." [3]



(a)

Capability	Configuration							
	FY88	I	II	III	IIIA	IIIB	IIIC	IV
Carrier Suitability	G	Y	Y	Y	Y	G	G	Y
Strike Mission	Y	Y	G	G	G	G	G	G
Fighter Mission	Y	Y	G	G	G	G	G	G
Maneuverability	R	R	R	R	R	R	Y	Y
Fire Control System	R	G	G	G	G	G	G	G
Survivability	Y	G	G	G	G	G	G	G
Growth Potential	Y	R	R	Y	Y	G	G	G
Weapon System Effectiveness	R	Y	Y	G	G	G	G	G
Non-Dimensional Cost Range REC	1.00	1.14	1.30	1.37	1.36	1.40	1.39	1.46
NR	–	1.00	2.22	2.88	2.77	2.89	2.78	3.44

G Good   Y Marginal   R Serious Concern

(b)

Figure 27: Design alternatives and evaluation criteria used during the conceptual design of the F/A-18 E/F Super Hornet [3]

Although serious for engineering problems, this issue is even more prominent in other fields such as fashion design or interior decoration where there may be no easily quantifiable criteria like weight or performance. In these cases, the burden to evaluate the merit of designs falls squarely on the shoulders of the analysts. This deficiency of available methods was one of the driving forces behind the development of Interactive Genetic Algorithms (IGAs), which provide optimization capability for problems that have unquantifiable objectives like maximization of beauty rather than the more traditional optimization objectives used in engineering. Other than the fact that fitness evaluation is performed by a human rather than a computer simulation, Interactive GAs are similar to conventional Genetic Algorithms.

Dawkins was one of the first to develop and use Interactive Evolutionary Computing (IEC), and he created a system that evolved artificial creatures known as biomorphs by relying on asexual reproduction and aesthetic evaluation. [34] Another important early work was that of Caldwell and Johnston, who designed an IGA system to assist crime victims in the creation of composite sketches. [23] Since then, the IEC community has grown and split into two largely distinct categories that have been coined "broad" and "narrow" by Takagi, the author of an excellent survey of IEC applications [149].

Narrow IGAs: Of the two types, narrow Interactive Genetic Algorithms are most similar to Dawkins' original work, because in applications of this category the analyst directly evaluates each population member based upon his or her subjective preference. This evaluation is typically performed using a graphical user interface, such as those found in Figure 28. These interfaces provide a slider, list box, or other input method that allows the human evaluator to provide a ranking for each concept. The subjective ranking is then used as the fitness value during the selection process. "Narrow" IEC methods have been applied to a diverse set of problems including music composition[12], floor plan layout[47], fashion design[84], and furniture design[10].

Broad IGAs: In "broad" IEC applications, the decision maker does not directly evaluate the designs but guides the evolutionary algorithm in other ways like updating

(a) User interface used to obtain input for 3-D model design [84]

(b) User interface used to obtain input for fashion design [115]

Figure 28: Examples of IEC user interfaces

the weights on an overall evaluation criterion as the search progresses. This type of IGA implementation gives the designer a degree of control over how the design space evolves with time, and is therefore most appropriate for problems with multiple conflicting objectives. Broad IEC techniques have been applied to problems such as nurse scheduling[76] and aircraft design[97][120].

Though there have been many applications of both types, Takagi has identified several difficulties with current IEC practice. The most important issue is that of human fatigue, because it can be very tedious to manually evaluate the large number of design alternatives required by IEC techniques. Several researchers have proposed methods including neural network or nearest neighbor prediction to ease the burden on the human operator, but the benefits of these approaches remain unclear. [13] Another promising method to reduce the workload is to limit the number of fitness levels available to choose from, though this increases quantization noise and may delay convergence in later generations. [150]

2.3.5.5 Conceptual Design Applications

Applications of evolutionary algorithms to conceptual design problems can be divided into two categories, described by Bentley as *evolutionary design optimization* and *conceptual evolutionary design.* [11] When performing *evolutionary design optimization*, the decision variables of an existing concept that are felt to be important or need improvement are parameterized by the analyst and then optimized using evolutionary search techniques. This was naturally one of the earliest applications of evolutionary search to design because it deals with the same type of problem as more traditional optimization methods. Several designers have applied this technique to evolve vehicles that exhibit considerably improved performance compared to the reference design. Rasheed [126] used a GA to minimize the takeoff mass of a reference supersonic aircraft, and achieved a reduction of 33% over the baseline concept, but the use of very simplified analyses limits the usefulness of the results. Bos[17] and Roth and Crossley[132] tackled similar problems with GAs but again found that the reliance on drastically simplified analyses led to suspicious outcomes.

Conceptual evolutionary design is a much broader application of evolutionary techniques than evolutionary design optimization because it aids in both system synthesis and parameter optimization. Applications of this type to aerospace design have been much less frequent than the previously discussed methods. Roth and Crossley's approach did include integer variables such as the number of engines or aisles thus allowing for designs of different morphologies, but the author does not view this as a true system synthesis problem because the number of design variables was very limited and no hierarchical synthesis variables were included. In [113], Mosher used Genetic Algorithms to optimize a satellite launch system. The problem included discrete variables such as material or battery type, but again the lack of a system hierarchy means that this cannot truly be considered conceptual evolutionary design.

Conceptual evolutionary design techniques have been more frequently applied to problems in other disciplines. As previously mentioned, Parmee applied the Structured Genetic Algorithm to the system design of a hydropower plant[120]. In [154], Thompson applied GAs to analog circuit design and found that the use of GAs helped to produce novel circuit concepts of high performance. Leger [91] used evolutionary algorithms to design innovative robot configurations of several different types that are capable of performing difficult tasks. Each of these applications found new and innovative ways of applying evolutionary search techniques to system design, but none fully addressed each of the barriers to MDO acceptance that were introduced in Section 1.3.

2.3.6 Ordinal Optimization

Ordinal Optimization is a method that seeks to find "good enough" or "satisficing" solutions rather than "optimal" solutions. [67] The method relies upon two basic tenets: order is much easier to determine than value and goal softening decreases the computational expense of finding good designs.

The first principle can be explained such that when presented with two alternatives A and B, it is much easier to determine whether A is better than B rather than A-B = ?. Even if estimates of the values of A and B are relatively uncertain, the chance of mis-ordering A and B is relatively small. The implications of this fact are that even relatively simple "back of the envelope" calculations or heuristics provide substantial utility when looking for "good enough" solutions early in a design project. As an example of this principle, an individual might have difficulty in estimating the length of an object by eye. However, if shown two objects, it is trivial to determine which one is longer.

Goal softening is directly related to the search for a set of "good enough" solutions rather than a single optimal outcome. When presented with possibly millions or billions of concepts in the first stages of a design program, the idea of selecting a single optimum is likely ridiculous. Ordinal optimization instead advocates finding a potentially good subset of solutions through the use of crude models, as shown in Figure 29. Even if these models are relatively inaccurate, the sample is likely to contain at least one truly high performance concept. Ho, the inventor of Ordinal Optimization, characterizes this redefinition as "hitting a truck with a shotgun" rather than "a speeding bullet with a speeding bullet."

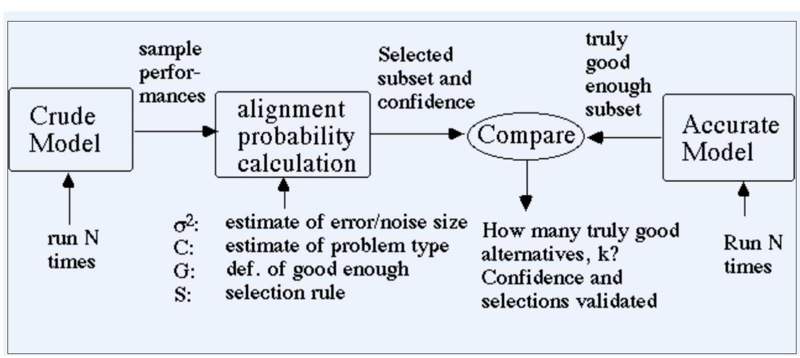


Figure 29: Outline of the Ordinal Optimization Process [119]

Ordinal optimization has obvious synergism with Genetic Algorithms for conceptual design applications because, as its creator points out, it justifies the use of speedy but possibly inaccurate simulation when the objective is to find a good reference configuration rather than to wring the last percentage of performance out of a design.

2.4 Multi-Objective Optimization

In engineering design, there are almost always multiple criteria that must be considered during the concept selection process. Objectives such as weight, cost, and speed must be balanced versus each other to find that "right mix" that will result in a successful program. Because most optimization techniques require a scalar objective function value, the analyst must aggregate these objectives in some fashion. Two largely distinct classes of multiobjective formulation methods known as *a priori* and *a posteriori* have been developed to aid in multi-objective problem definition.

A priori: This class of multi-objective optimization methods seek to create an aggregate objective function that expresses the preferences of the decision makers. Once formulated, this scalar value can be optimized using traditional techniques to find a single "best" solution to the multi-objective problem.

A posteriori: This class of multi-objective optimization methods differ significantly from *a priori* methods because methods of this type require no preference or goal information from the decision maker before performing an optimization run. Instead, these methods seek to identify a *set* of solutions which are all *Pareto optimal.*

Definition (Pareto dominance): Solution A is said to *Pareto dominate* solution B if every objective of A is at least as good as B, and at least one objective of A is better than B:

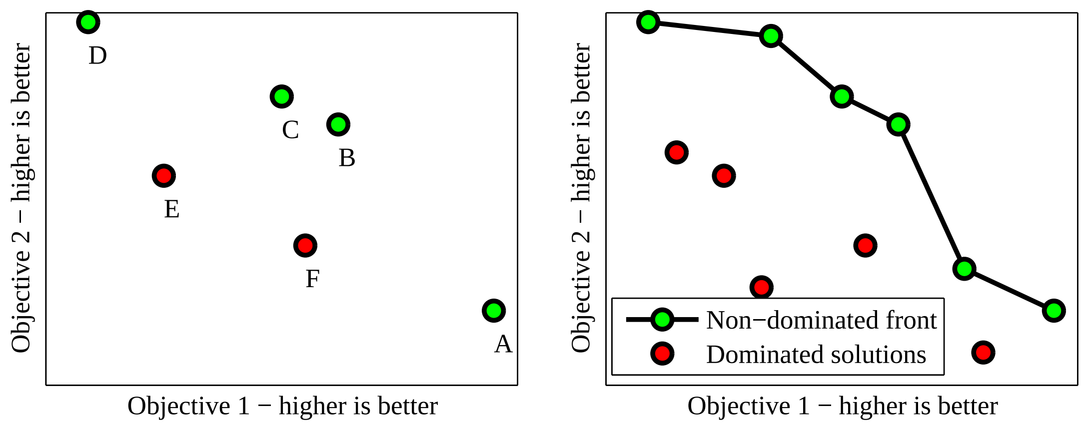
$$\forall_i \in \{1:n\} : f_i(a) \leq f_i(b) \wedge \exists_j \in \{1:n\} : f_j(a) < f_j(b) \qquad (6)$$

Definition (Pareto optimal set): A set of solutions $\bar{P}$ is a *Pareto optimal* set if there is no solution in the search space that Pareto dominates any member of $\bar{P}$.

Definition (Pareto frontier): The set of objective values $\bar{F}$ corresponding to the Pareto optimal set $\bar{P}$ is known as the *Pareto frontier*.

The concepts of Pareto dominance and optimality can perhaps be best understood through a visual inspection of Figure 30. In Figure 30(a), points A-D are non-dominated with respect to the solutions shown because none is better than another with respect to both of the objectives plotted in the figure. Points E and F , however, are dominated solutions because there are other designs present that have better performance in both objectives.

Once a set of Pareto optimal solutions has been found, these results are typically displayed as a Pareto frontier, which graphically depicts the tradeoff between the problem's objectives. This type of result can be a very powerful tool for decision makers as it can assist in understanding the relationship between objectives and therefore help to establish realistic requirements for the design. (Figure 30(b)).



(a) An example of the non-dominance concept (b) A notional Pareto frontier

Figure 30: Illustrations of the concept of Pareto optimality

In conceptual design problems, one is often interested in not only viewing the relationship between requirements but also in determining which configuration makes the most sense in a given region of the requirements hyperspace. The s-Pareto frontier [100] is a tool for the visualization of this type of result, and clearly shows the relationship between objectives and configurations. In the notional example of Figure 31, concepts 1 and 2 make up the s-Pareto front, while concept 3 is completely dominated because it is inferior with regard to both objectives.

After the Pareto frontier has been located, the decision makers must explore the results and select a best compromise design. This may be done by visual inspection or with the aid of Multi-Attribute Decision Methods such as TOPSIS. [74]

2.4.1 A Priori-based aggregation methods

2.4.1.1 Weighted Sum Method

The most commonly used and intuitive method for converting a multi-objective problem into a single-objective problem is the weighted sum:

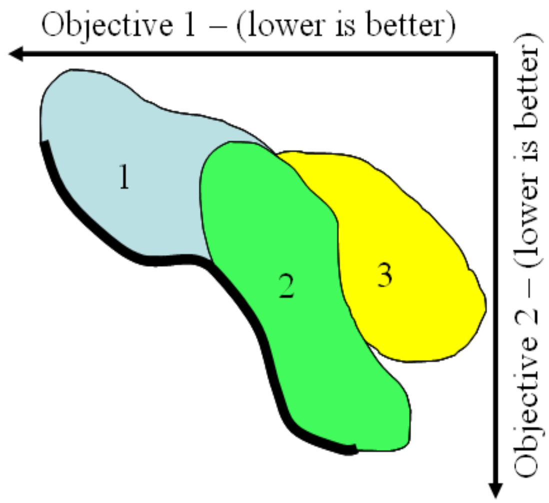


Figure 31: The s-Pareto frontier

$$F(x) = \sum_{m=1}^{M} w_m f_m(x) \tag{7}$$

where M is the number of objectives, f_m is the m^{th} objective function, and w_m is the weight of the m^{th} objective. Although not strictly required, in practice each m^{th} objective function is often normalized so that the weights w_m are more directly related to decision-maker preference.

Although simple to formulate, the weighted sum method has several shortcomings. One issue is that "optimum" designs obtained using a weighted sum technique may not best meet the users' goals because it is difficult to numerically quantify how important objectives are relative to each other. [32] The user may therefore be forced to re-run the algorithm with new weights in a trial-by-error fashion. Another problem with the weighted sum technique is that it can be mathematically proven that weighted sum optima will never be located on non-convex regions of a Pareto frontier. [38]

2.4.1.2 Weighted Metric Methods

Weighted metric methods recast the aggregation problem by seeking to minimize the l_p distance to the ideal solution z^*:

$$l_p(x) = \left(\sum_{m=1}^{M} w_m |f_m(x) - z_m^*|^p\right)^{1/p} \tag{8}$$

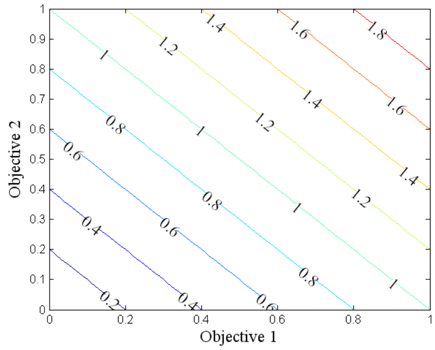
where M is the number of objectives, f_m is the m^{th} objective function, and w_m is the weight of the m^{th} objective.

The parameter p controls the order of the relationship, and several specific categories of weighted metrics have been defined. When $p = 1$, the above equation yields the Manhattan metric, which is equivalent to the weighted sum. Setting $p = 2$ yields the Euclidean metric, and p=∞ is the Tchebychev metric. One advantage of the Tchebychev metric is that it can be shown that even non-convex Pareto optimal solutions can be found by varying the weights w_m in Equation 8, but the Tchebychev problem is not continuous and therefore poses difficulties for classical optimization techniques.

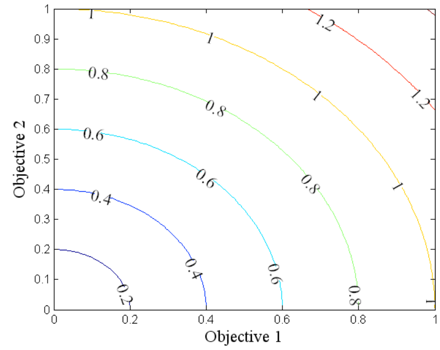
2.4.1.3 Goal Programming

Goal programming is a problem formulation method that seeks to find an optimal compromise solution by minimizing the deviation from a set of supplied target objectives. [75] In the goal programming approach, each objective is categorized according to four criteria:

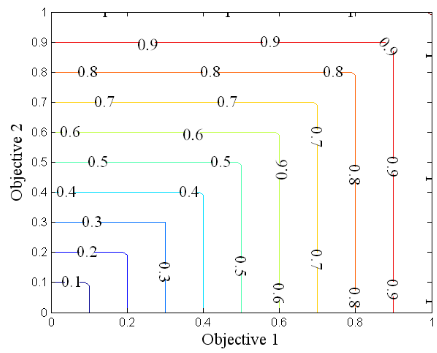
1. Less than or equal to
2. Greater than or equal to



(a) $p = 1$



(b) $p = 2$



(c) $p = \infty$

Figure 32: Contours of the l_p metric for different values of p

3. Equal to
4. Within a range

The problem is then recast as the minimization of the weighted difference between the actual and goal values. In this sense, goal programming is similar to a weighted l_p metric problem with $p = 2$, except for the fact the reference vector is a user supplied set of desired objectives rather than the ideal solution.

2.4.1.4 Utility Theory and Physical Programming

Utility theory is a powerful technique for use in multi-criteria problem formulation that seeks to create a non-linear value function that completely expresses the designer's preferences. According to its definition, a concept of higher utility will always be preferable to one of lower utility. Additionally, two designs of equal utility will be equally preferable. In the case where a utility function is available, this technique is likely the best possible method for solving multi-objective problems. Unfortunately, the formulation of such a function can be non-trivial and error-prone.

Physical Programming [108] is an intuitive method closely related to goal programming that can be used to create utility functions through the use of a set of soft classes that allow the decision maker to more explicitly express his or her wishes. (Figure 33) By classifying responses in an intuitive manner with labels such as "highly desirable" or "acceptable", the designer is able to quickly create single attribute utility functions which are then additively aggregated to form a scalar objective function.

Several researchers have found that Physical Programming yields improved results compared to other aggregation methods, but a remaining weakness is that weights must still be assigned to each of the class functions used to form the objective.

2.4.1.5 ϵ-Constraint Method

One final way frequently used to address multi-objective problems is to set all but one of the objectives as constraints and then use a constrained optimization technique such as Sequential Quadratic Programming[14] or the Method of Feasible Directions[167] to minimize the remaining objective subject to the specified constraints. The problems with the constrained approach include the fact that it does not give the designer any insight into how the requirements relate to each other. This means that problems similar to those encountered during the SST program discussed in the motivation section are likely to be encountered. Also, because the tools used at this stage of the

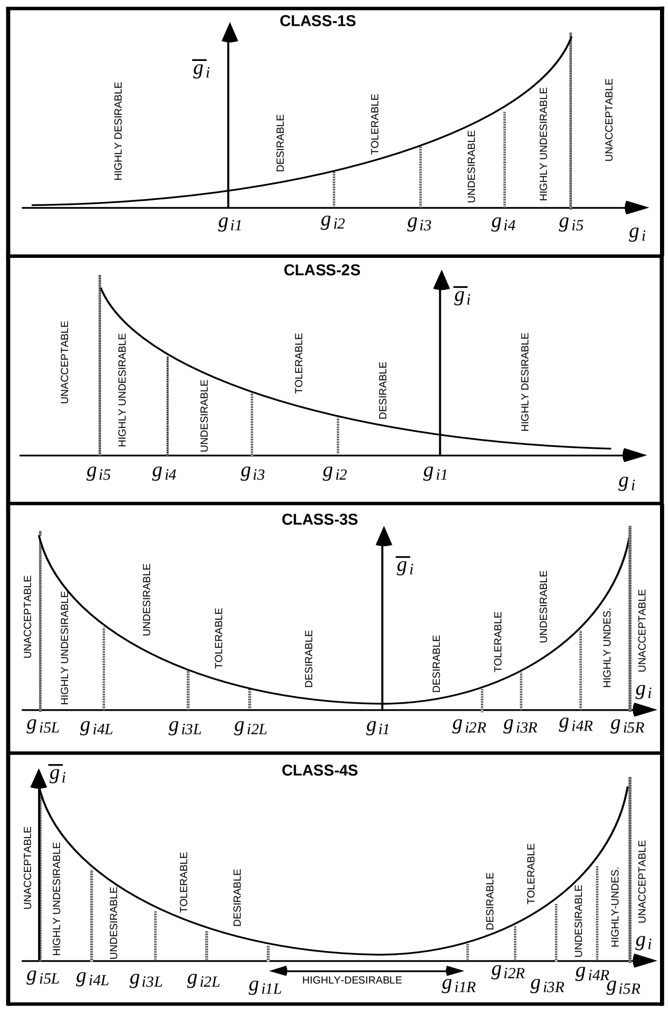


Figure 33: Physical Programming class functions [108]

design process are relatively inaccurate, constraints are often "fuzzy" - a deviation of a few percent in a value may be insignificant to the decision maker, but if that metric is constrained it may lead to unsatisfactory answers.

2.4.2 A Posteriori-based Methods

2.4.2.1 Multiple applications of a priori methods

A large number of optimization methods have been proposed for finding the Pareto front. The earliest and most obvious of ways to locate this set of solutions is the application of successive optimization runs using one of the a priori aggregation methods and different weight vectors. By varying the coefficients used to calculate the weighted sum or other metric, each optimization run will converge to a different location on the Pareto front. Unfortunately, there are several difficulties with this approach. The primary problem is that the need for multiple optimization runs may cause the computational expense of the problem to quickly grow out of control. Another issue is the fact that it can be mathematically proven that a well distributed set of weight vectors does not necessarily result in a good distribution of solutions along the Pareto frontier, as illustrated in Figure 34.

This latter difficulty can be addressed by the using the ϵ-constraint method, but the need for multiple runs may be prohibitive, especially on problems with more than two objectives.

2.4.2.2 Multiple Objective Genetic Algorithms

One of the most obvious differences between classical and evolutionary optimization techniques is the fact that Genetic Algorithms operate on a population of individuals.

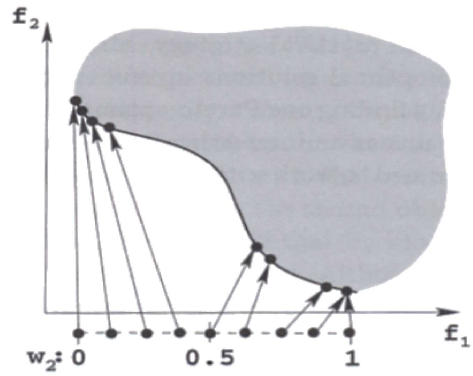


Figure 34: Poor Pareto frontier representation obtained using the weighted sum approach [38]

This fact gives Genetic Algorithms a distinct advantage for multiobjective applications because the goal of a posteriori methods is a *set* of solutions rather than a single point. Over the last twenty years a number of researchers have recognized this fact, and developed increasingly capable Multiple Objective Genetic Algorithms (MOGAs).

Vector Evaluated Genetic Algorithm This suitability of Genetic Algorithms to multi-objective problems was first recognized by Schaffer, who developed the Vector Evaluated Genetic Algorithm (VEGA) in 1984. [134] VEGA solves the multi-objective problem by dividing the population into a number of subpopulations equal to the number of objectives. Within each subpopulation, the selection operator is performed using the i^{th} objective function. This algorithm is very easy to implement, but it has several disadvantages. The most important of these is the fact that VEGA will tend to emphasize "champion" solutions that excel in one particular objective, while ignoring "compromise" solutions that may be of more interest to the decision maker.

Non-dominated sorting MOGAs

In [60], Goldberg proposed a new method of multi-objective fitness assignment known as non-dominated sorting. This procedure of fitness assignment, illustrated in Figure 35, calculates the rank of each individual according to how many other fronts dominate the front that it is a member of. Goldberg also suggested the use of a niching strategy that differentiates among solutions of the same non-domination rank and prefers solutions in less densely populated regions of the design space, thereby promoting diversity.

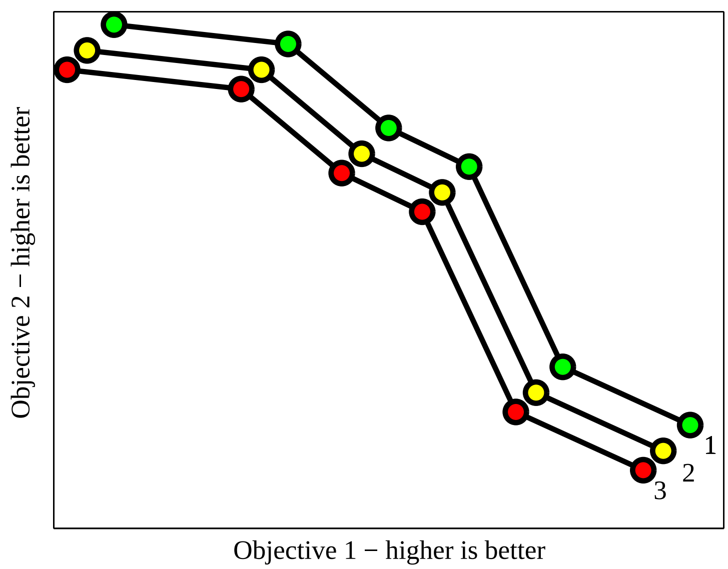


Figure 35: The principle of non-dominated sorting: solutions belonging to dominated fronts are assigned successively inferior fitness values

As a result of these suggestions, several improved multiple objective Genetic Algorithms were developed in the 1990s, including the Niched Pareto Genetic Algorithm [70], the Multiple Objective Genetic Algorithm [56], and the Non-dominated Sorting Genetic Algorithm [144]. Each of these methods was shown to perform significantly better than the original VEGA. The interested reader is referred to [38] for a comprehensive review of the relative merits of each of these methods, but for completeness the NSGA procedure will be described here.

The Non-dominated Sorting Genetic Algorithm, or NSGA, is similar to the standard genetic algorithm apart from its method of fitness assignment. (Figure 36) Under the NSGA, all non-dominated solutions are located, and assigned a large “dummy” fitness value. These dummy fitness values are then degraded through the use of sharing, where the original assigned dummy fitness value is divided by the number of other members in the front that are within σ_{share} distance of the individual. The non-dominated solutions are temporarily ignored, and the procedure is repeated for the remaining individuals, using a smaller assigned dummy fitness value at each step. After all population members have been assigned dummy fitness values, the algorithm proceeds in the same fashion as the standard Genetic Algorithm, with selection preference given to designs with large dummy fitness values.

Elitist MOGAs

Several criticisms of the Non-dominated sorting class of MOGAs emerged in the late 1990s. The two most prominent issues identified were the fact that prior methods did not use an elitist strategy, and the requirement for the analyst to specify the σ_{share} used to maintain diversity. [39] In response, a new class of algorithms including the Non Dominated Sorting Genetic Algorithm II (NSGA-II)[39] and the Strength Pareto Evolutionary Algorithm 2 (SPEA2)[164] were

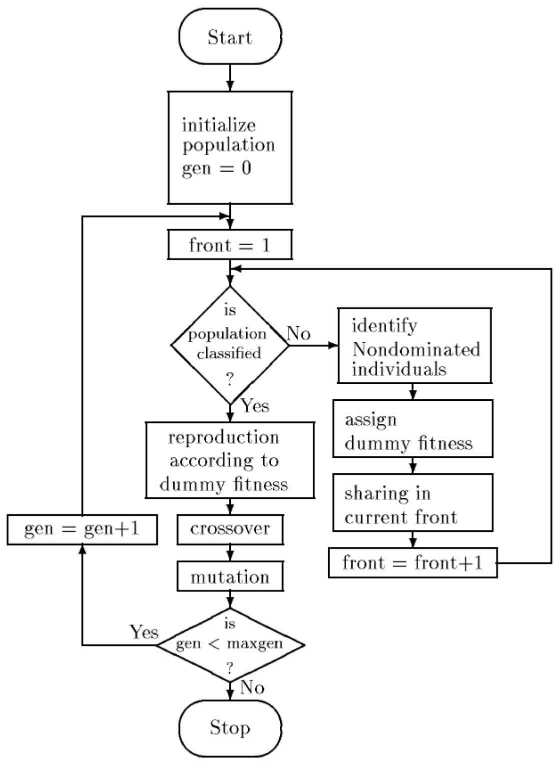


Figure 36: The Non-dominated Sorting Genetic Algorithm (NSGA) [144]

developed that perform significantly better than their predecessors. These two methods operate using many of the same principles, but several reports have found that the SPEA2 algorithm performs slightly better than the NSGA-II because of its diversity preservation routine, and therefore the SPEA2 will be used as an example of a modern, elitist Multiple Objective Genetic Algorithm.

The SPEA2 algorithm is similar to the standard genetic algorithm apart from its use of environmental selection and its fitness calculation procedure. The algorithm was developed based upon its creators experiences with the earlier Strength Pareto Evolutionary Algorithm (SPEA). [165] Although the SPEA performed well on a number of test problems, it encountered difficulties if there happened to be only one Pareto optimal solution in a given population, and would on occasion lose boundary solutions. The new environmental selection and fitness assignment routines associated with SPEA2 remedy these issues.

Fitness calculation in the SPEA2 is a multi-step procedure, and is found using the union of two sets of points: those in the current population P_t, and those from an archive of solutions $\bar{P}_t$. First, the strength $S(i)$ of each population member must be calculated, which is equal to the number of solutions that are dominated by solution i. The raw fitness of each solution $R(i)$ is then calculated through the summation of the strength values S for designs that dominate solution i.

$$R(i) = \sum_{j \succ i} S(j) \tag{9}$$

In addition to the raw fitness, a distance measure $D(i)$ is calculated that serves to prefer solutions in sparsely populated regions:

$$D(i) = \frac{1}{\sigma_i^k + 2} \tag{10}$$

where $k = \sqrt{N}$, N is the size of the evaluated set, and σ_i^k is the distance (in objective space) from point i to its k_{th} nearest neighbor. Finally, fitness is calculated by summing R and D.

The environmental selection step is used to ensure elitism. If the number of non-dominated solutions is less than the size of the archive, the archive is simply filled with the designs with the best fitness. In the case where there are more non-dominated solutions than can fit in the archive, a truncation operator is applied that iteratively removes the most crowded solutions.

2.4.2.3 MOEA Challenges

Although MOEA methods have improved greatly since their inception, several limitations remain, especially for problems with a large number of objectives. Deb [38] investigated the relationship between the number of objectives and the proportion of a random population that is non dominated, as a function of population size. He found that as the number of objectives grows, nearly all solutions become non-dominated, and would therefore have equal fitness values. (Figure 38(a))

This presents a serious difficulty because no solution will have any selection advantage. Without selection pressure, Genetic Algorithms tend to stagnate and degenerate to a random search. As illustrated in Figure 38(b), large population sizes can decrease

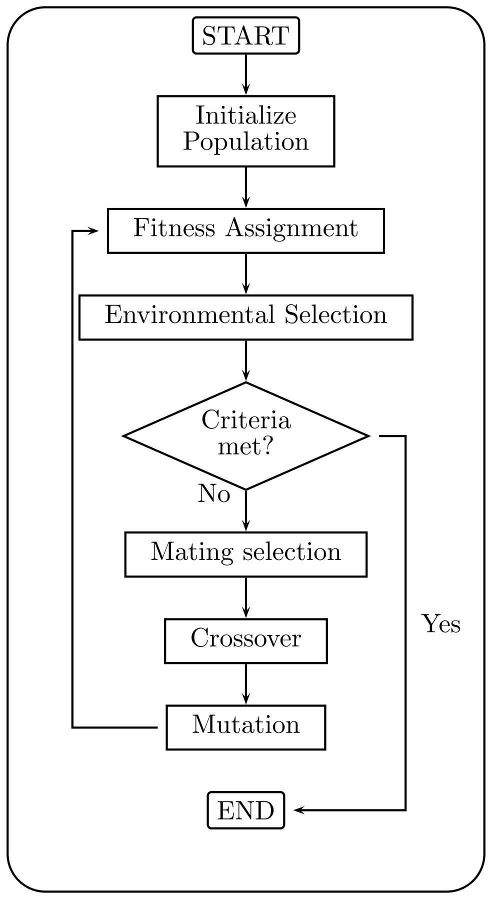
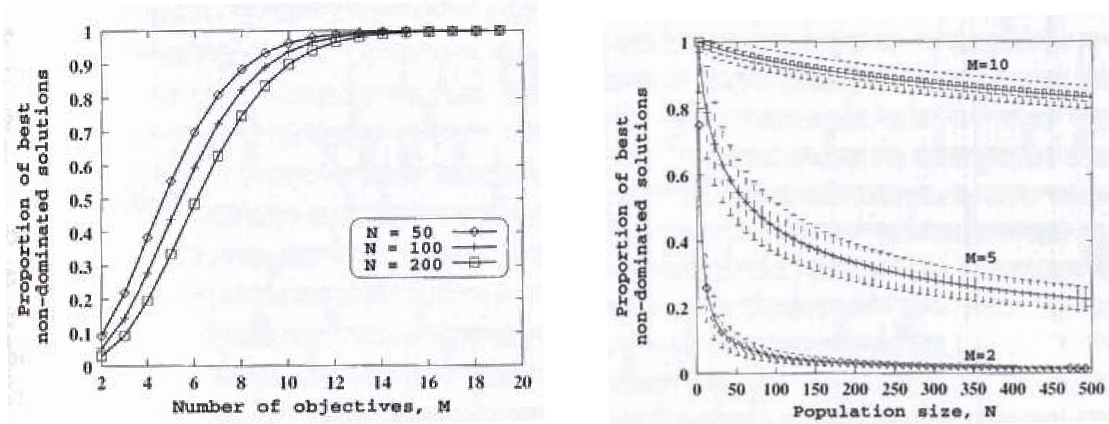


Figure 37: The SPEA2 Algorithm



(a) Proportion of non-dominated solutions as a function of the number of objectives

(b) Proportion of non-dominated solutions as a function of population size

Figure 38: Impact of the number of objectives on the proportion of non-dominated solutions in a population [38]

the proportion of non-dominated solutions and improve algorithm performance, but only at a large computational expense. For problems with more than approximately five objectives, nearly any practical population size will result in insufficient selection pressure to maintain algorithm performance.

The problem of locating the s-Pareto front is even more difficult because multiple configurations are being investigated. To date, determination of the s-Pareto front has required individual evolutionary search runs for each concept under consideration. This results in unacceptable computational expense for problems with a large number of solution alternatives.

2.4.2.4 *Visualization of multiple objective optimization results*

One of the main attractions of *a posteriori* optimization methods is the knowledge that can be obtained through examination of the resulting tradeoff information. For problems with two or possibly three objectives, the resulting Pareto surface will be easy to understand. However, visualizing and interpreting the results of a Pareto hyper-surface with more than three objectives remains a challenge. Some of the most common methods for interpreting multi-dimensional tradeoff information are the scatter-plot and value-path methods.

Scatter-plot method: The scatter-plot method [106] displays Pareto hyper-surface information by plotting all $\binom{M}{2}$ possible combinations of objectives in an M by M matrix. Figure 79 presents an example of a scatter-plot matrix for a problem with five objectives. This representation allows the decision makers to view all possible two-objective tradeoffs, but reveals little about interactions between more than two objectives. It may therefore be difficult to make accurate conclusions about complex objective relationships that are often present for engineering problems.

Value-path method: The Value-path method [57] is a popular technique for the visualization of multidimensional data sets. On a value-path diagram, M ticks are created along the horizontal axis of the diagram. (Figure 40) Each of the M objectives is normalized and plotted on the vertical axis, and the values of each objective corresponding to a single solution are joined by a line.

Examination of the value-path plot can reveal a large amount of information about the trade-off between objectives. The plot can also be used to locate "compromise" solutions. One issue is that while the value-path method scales well with the number of objectives, it can become very confusing when a large number of Pareto-optimal points are displayed.

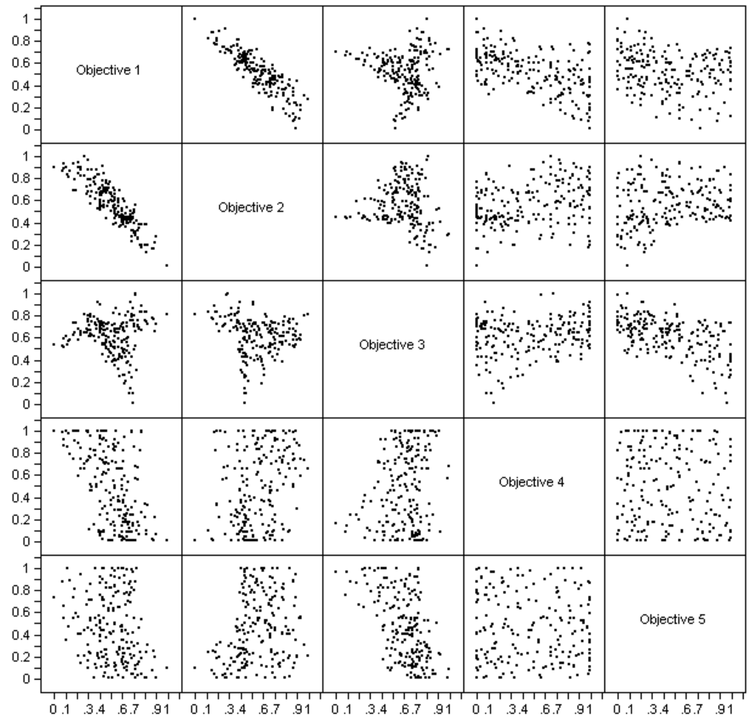


Figure 39: Scatterplot representation of a Pareto hyper-surface with five dimensions

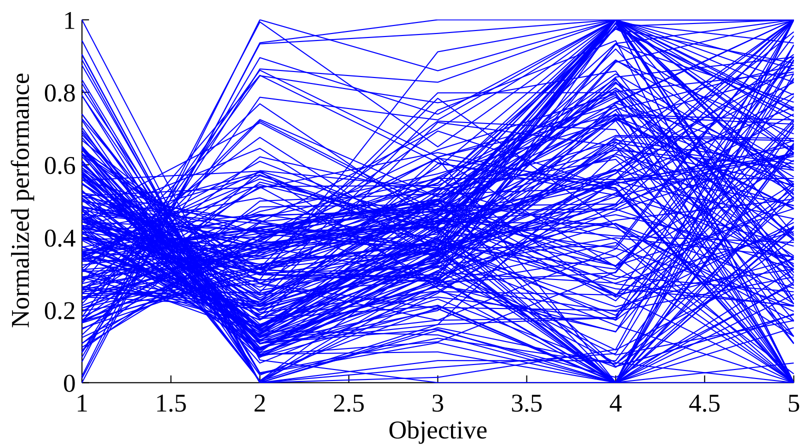


Figure 40: Value-path representation of a Pareto hyper-surface with five dimensions

2.5 Summary

A review of the relevant literature has revealed that while there is no existing method that addresses every cited issue with the design of revolutionary vehicles, there are several aspects of existing techniques that could be used as building blocks for a new, more capable design method. Morphological Analysis was found to be a powerful tool that promotes innovation by efficiently displaying the available options in the matrix of alternatives, but the lack of an available method to search this matrix is a significant problem. Evolutionary algorithms are a natural choice to assist in this search, but the algorithms available in literature were found to be inefficient for hierarchical design problems with many objectives and alternatives. Also, several promising multiobjective optimization methods were discussed, but scaling and visualization limitations have prevented their widespread application to problems with more than a small number of objectives. Finally, if the results of a new method are to be useful, all important criteria, both quantifiable and unquantifiable, need to be accounted for in some fashion or the resulting designs will tend to be biased towards undesirable regions of the concept space and ultimately prove to be unusable.

CHAPTER III

HYPOTHESIS AND RESEARCH QUESTIONS

A review of available design methods has shown that a need currently exists for a technique that is capable of efficiently searching the Matrix of Alternatives for promising concepts, and of providing insight into the relationships between a vehicle's requirements and its preferred attributes for creative design problems. However, the literature review also revealed that there are several obstacles to the creation of such a method. These observations resulted in the following research questions.

3.1 Research questions

Question 1: *How can engineering judgment and expertise be best combined with numerical analysis and optimization techniques to improve the conceptual design process?*

It is clear that many aspects of the design process require human judgement and expert input. At the same time, computer simulations and analysis can provide valuable insight into design problems, especially for creative design problems with which the designer has little experience. Combining the two methods of analysis is a challenge that must be addressed by this work.

Question 2: *Is it possible to rigorously search the Matrix of Alternatives for promising concepts without individually optimizing every possible alternative?*

Most methods for concept evaluation available in the literature require the analyst to individually optimize each alternative in order to perform a meaningful comparison. This may be feasible when there are only a handful of alternatives, but is computationally prohibitive in the case where there are hundreds or thousands of promising configurations. A method that is able to efficiently search the concept matrix by taking advantage of hierarchical decomposition would enable a more thorough exploration of system concepts and thereby increase designer freedom and knowledge.

Question 3: *Is there a computationally feasible way to use physics-based analysis tools in addition to the historically-based design guidelines commonly used during the concept generation phase of routine design programs?*

A number of multi-objective optimization methods are available in the literature, but these have difficulty in solving real-world engineering problems that may have more than two or three objectives. The fact that the relationship between conflicting objectives is often unknown when dealing with creative design problems requires the use of a method that will provide insight into this behavior.

3.2 Hypothesis

A method that allows designers to use physics-based analysis tools in concert with expert engineering judgement during the requirements and concept exploration stages of the design process will enable a more thorough *examination of the combinatorial system alternatives matrix than is possible using traditional design practice. A method that meets these criteria can be obtained by combining the capabilities of Interactive and Multiobjective Evolutionary Computation, thereby facilitating the application of relatively sophisticated analysis methods to the task of concept exploration.*

The analysis and optimization methods developed in the past twenty years have resulted in the attainment of a large portion of the "paradigm shift" shown in Figure 3. However, the majority of work to date has focused on improving upon sizing and design optimization rather than improving the earlier steps of design synthesis and requirements analysis. Although some methods have been developed to assist in the synthesis process, none has satisfactorily addressed each of the difficulties encountered during revolutionary system design. Because of this, much of the paradigm shift associated with the earliest stages of the engineering process remains unrealized. The hypothesis of this work states that if a method could be developed that enables the application of the tools commonly used by engineers in the later stages of the design process to the concept and requirements exploration phase, it would result in the attainment of a significant portion of the paradigm shift that has yet to be realized, and therefore lead to better designs.

3.3 Problem Statement

The objective of this thesis is to develop a method that will enable the paradigm shift that has already occurred later in the design cycle to be propagated upstream to the requirements and concept selection phase. The method will be developed and demonstrated using aircraft design as a test case, but the fundamental process should be applicable to a broader class of system design problems.

The method must be capable of handling both qualitative and quantitative evaluation criteria. This implies that a way to efficiently combine these different types of metrics must be developed. Expert input must be gathered in an intuitive and user-friendly manner, and methods must be developed that will reduce the computational burden and the resulting wait time for the user.

In order to demonstrate the method on a relevant aerospace design problem, an analysis environment must be developed capable of evaluating the quantitative performance criteria such as vehicle weight, range, and environmental impact with sufficient accuracy and low computational overhead. The environment must be flexible enough to evaluate designs of different morphologies and modular so that as new analyses become available they may be quickly integrated into the system. Once these issues have been successfully addressed, the resulting method will be a useful tool capable of providing the engineer with much greater insight into the relationship between configuration and requirements for revolutionary conceptual design problems.

CHAPTER IV

SOLUTION APPROACHES AND METHOD DEVELOPMENT

A review of the literature has revealed the need for a method that gives design engineers greater insight into the impact of configuration and requirement decisions during the concept formulation stage of revolutionary design problems. This chapter describes the work performed in response to the research questions that have been identified as critical to the successful creation of such a method.

4.1 Hybrid qualitative/quantitative Interactive Genetic Algorithms (Question 1)

> *How can engineering judgment and expertise be best combined with numerical analysis and optimization techniques to improve the conceptual design process?*

In an ideal conceptual design environment, one would be able to obtain or create analytical models capable of accurately predicting all vehicle attributes. Within this environment, each disciplinary code would be linked together so that the impact of the design parameters on each attribute could be readily calculated for a large number of system architectures. Additionally, different levels of fidelity would be available, enabling the analyst to trade accuracy for computational expense as appropriate.

(Figure 41)

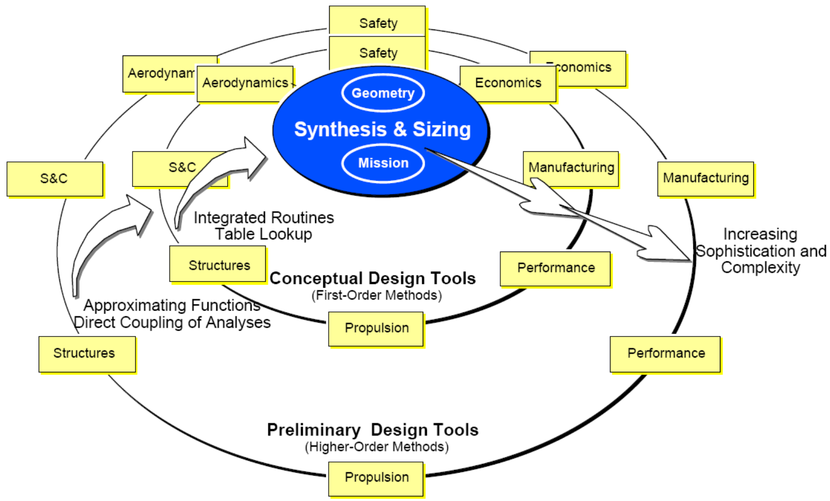


Figure 41: An integrated modeling and simulation environment with different levels of fidelity [102]

In reality, a complete and flexible environment such as the one depicted in Figure 41 does not currently exist, and may not for a significant period of time in the future. In the case of aircraft design, tools capable of accurately predicting weight, aerodynamic, and propulsion-related performance are readily available, but quantification of other critical parameters including aero-elastic phenomena and manufacturability remains difficult. (Figure 42) The result of this fidelity imbalance is that when the tools commonly used in aircraft design are linked with an optimizer, the resulting designs tend to be biased to perform well in the areas that are quantified with the greatest accuracy while ignoring other critical criteria that are not rigorously accounted for. This tendency is evident in the solutions that emerged from several recent supersonic business jet design studies, shown in Figure 43.

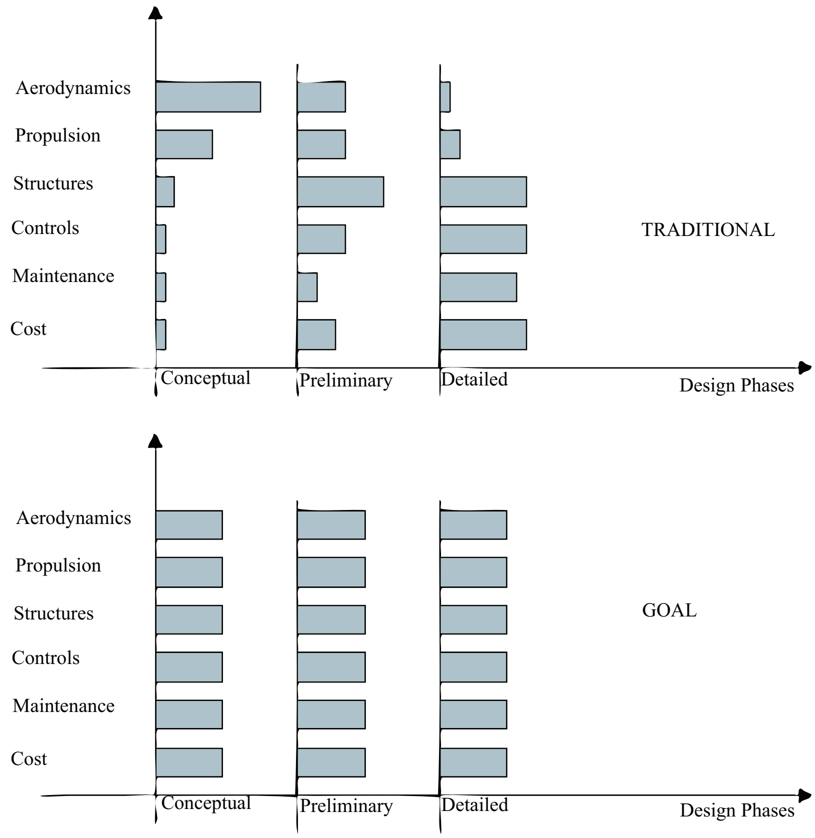
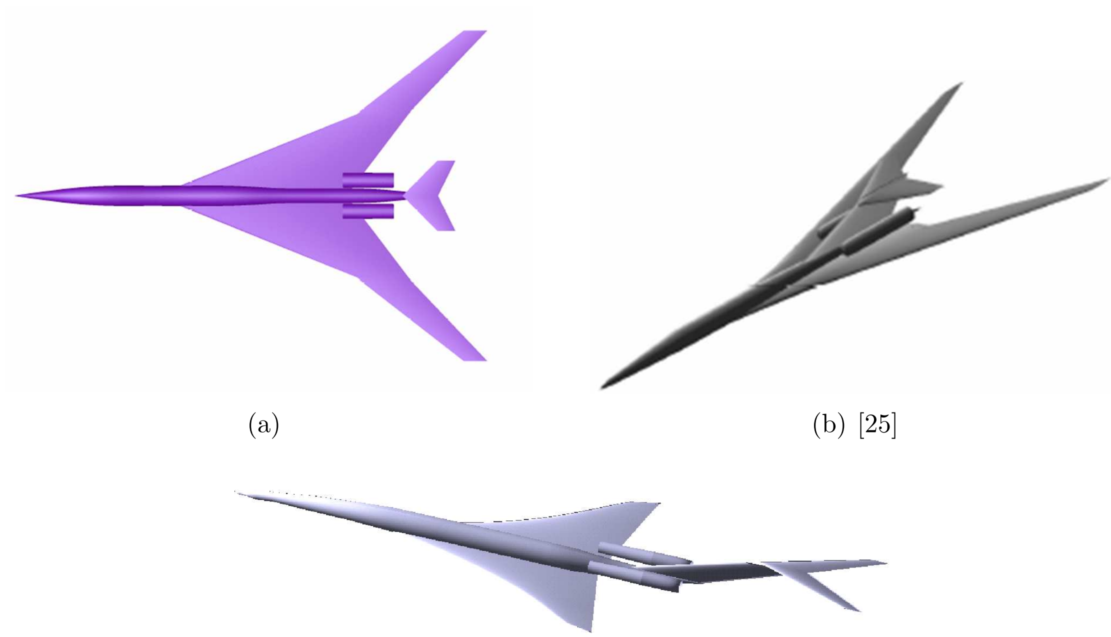


Figure 42: Distribution of effort in the aircraft design process [135]

The most typical approach used to prevent an optimizer from exploiting deficiencies in a simulation environment is to limit the decision variable ranges from which the optimizer is allowed to choose via tight side constraints. This approach works well if there are no significant interaction terms present in the model, but when interactions are present the technique is inadequate because it forces the designer to exclude a large amount of potentially feasible search space, thereby preventing the designers from locating the best solutions. As an example, in [20] a supersonic business jet was optimized using response surface equations. The authors found that the weight analysis methods used in the study were being exploited by the optimization routine,



(a) (b) [25]

(c) [97]

Figure 43: "Optimized" supersonic business jet configurations from the literature

and tightened the side constraints on the problem to prevent the selection of these infeasible designs. Although the resulting "optimized" configuration was found to be realistic, it was incapable of meeting all design requirements. These results highlight the fact that while the ultimate research goal may be to formulate methods capable of accurately quantifying these characteristics and thereby prevent exploitation, in the meantime it is advantageous to pursue other methods of assessing their impact.

4.1.1 Method formulation: The Hybrid Interactive Genetic Algorithm

The Interactive Genetic Algorithms that have been developed and applied by members of the computer art, fashion, and other creative communities have shown great promise for the optimization of difficult-to-quantify objectives. IGAs have been successfully applied to diverse problems including floor-plan layout and music composition.

Inspired by these applications, this work proposes to create a hybrid qualitative/quantitative Interactive Genetic Algorithm that combines the principles of Multi-objective and Interactive Genetic Algorithms. By relying upon both numerical simulation and expert input to evaluate the quality of a design concept, the method can be used to resolve the tradeoff between best performance as calculated by the computer and subjective satisfaction with a concept as indicated by the human analyst. Once found, this tradeoff can be represented as a Pareto front as depicted in Figure 44 and used to examine the relationship between the two goals. The successful implementation of the new method was determined to be largely dependant upon the resolution of several issues identified with traditional IGAs: those of human fatigue and user-friendly GUI design.

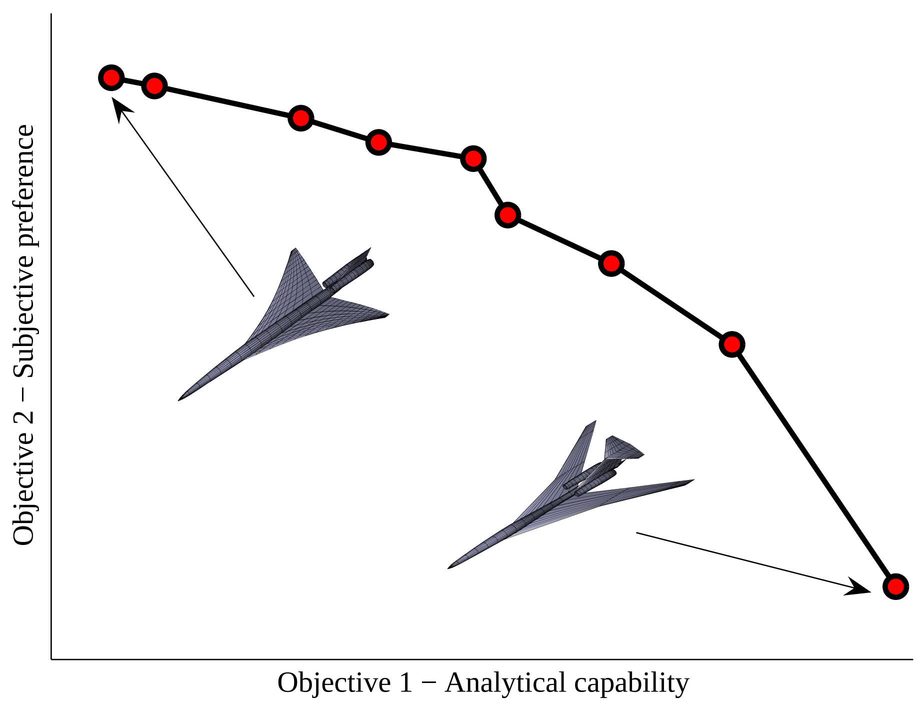


Figure 44: Goal of the hybrid method: resolution of the tradeoff between optimum numerical performance and user satisfaction with the concept

4.1.1.1 Interactive Genetic Algorithm GUI design

Perhaps the simplest way to minimize the effort required to evaluate concepts in an Interactive Genetic Algorithm is to ensure that the graphical user interface used to query the decision-maker is easy to understand and operate. Based upon the GUI's used in previous IGA applications, an interface (Figure 50) was created in the MATLAB environment capable of displaying a visual representation of the concepts evaluated by the algorithm.

Research described in [150] indicates that the burden on the human evaluator can be reduced by limiting the number of fitness values from which to choose. Based upon these findings, five fitness values were coded into the GUI: "Best", "Good", "Acceptable", "Poor", and "Bad". In addition to the list-box used to obtain user-input fitness values, each concept has a button labelled "Edit" that can be used to view and/or edit the values of the design variables. When depressed, this button brings up an additional window (Figure 45) that displays additional information about the concept and allows the decision-maker to make changes to the configuration, thereby embedding knowledge into the algorithm. Although this feature can accelerate convergence, frequent use may in fact slow the algorithm down because the engineer may become preoccupied with "fiddling" with the concept rather than evaluating the optimizer-generated designs.

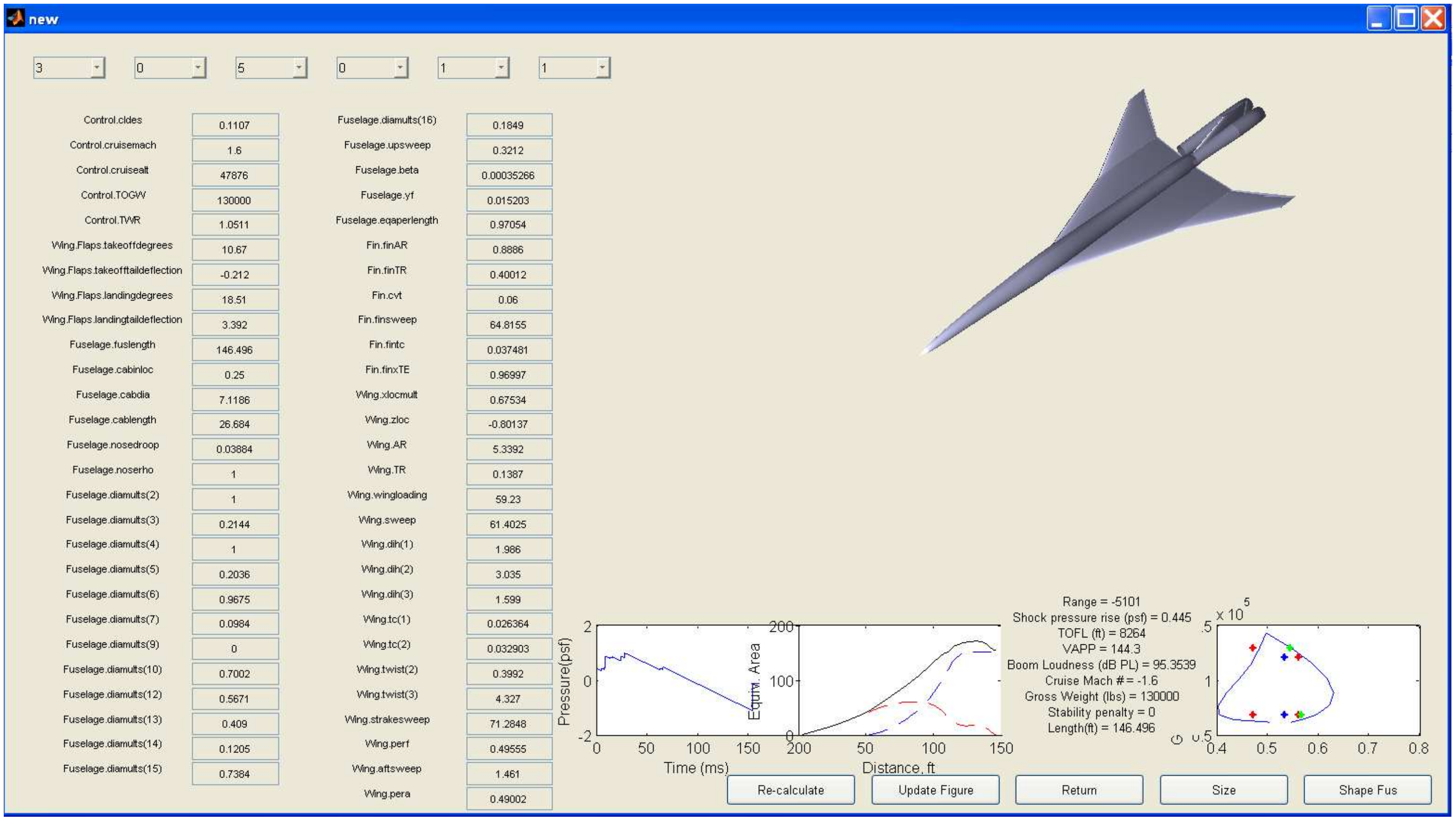


Figure 45: Graphical user interface used to view configuration details and embed knowledge on-line

4.1.1.2 Reduction of burden on the human IEC evaluator

Several studies have identified the primary problem with IEC techniques as that of human fatigue. [149][150] It was found that typical users could evaluate approximately twenty generations containing twenty individuals each before becoming tired. After this point, the human evaluator becomes disinterested and provides increasingly unreliable fitness estimates.

Non-interactive genetic algorithms frequently require more than 10,000 function evaluations to locate high-performance solutions. The limit of 400 subjective function evaluations is therefore especially constraining for this research into hybrid qualitative/quantitative GAs. Because of this, several techniques to accelerate algorithm convergence have been integrated into the present method.

Acceleration of Genetic Algorithms using "Injection Islands": An example of a successful attempt to improve the convergence rate of GAs is Eby's injection-island GA, described in [45]. This work employed variable fidelity analyses, using only the most accurate analysis for the ultimate fitness evaluation, while the low fidelity calculations were used to find promising candidates that were injected into the high fidelity analysis' population. (Figure 46) The method was demonstrated using a flywheel optimization problem, and was shown to improve convergence rate by nearly an order of magnitude compared to the classical single-population GA.

Although simplified analyses were used to predict fitness on the low order islands in the application described in [45], it would be possible to extend this concept to Interactive Genetic Algorithms that rely upon human-based fitness if it was possible

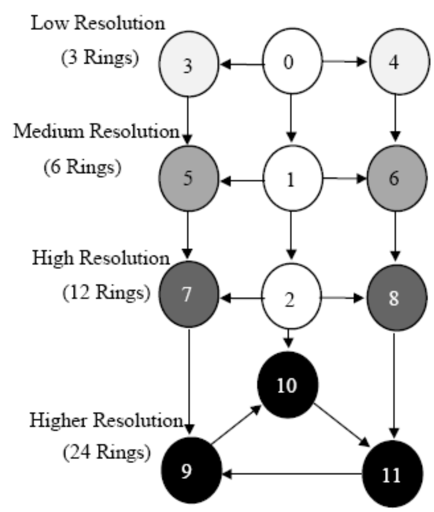


Figure 46: Injection island GA topology used for the design of flywheels [45]

to develop a method capable of approximating the human evaluator's reaction to a concept.

Prediction of subjective ranking using K-nearest neighbor interpolation: One solution to the IEC function limitation problem is to develop a method capable of predicting what a user's subjective evaluation will be without actually querying the user. In [117], Ohsaki investigated the effectiveness of two such methods, Neural Networks and K-nearest neighbor interpolation. He found that the use of a weighted nearest neighbor prediction scheme outperformed Neural Networks. Additionally, data indicated that the best results were obtained when only the individuals from the previous generation were used for prediction of the new population's subjective fitness.

The prediction scheme used in [117] is based upon a weighted average of the fitness values found using the K closest individuals in the parameter space. The fitness value of a new value P_{new} is predicted by generating K weighting distances corresponding to the K closest individuals and then taking the weighted average of the reference solutions' fitness values. This procedure is illustrated in Figure 47.

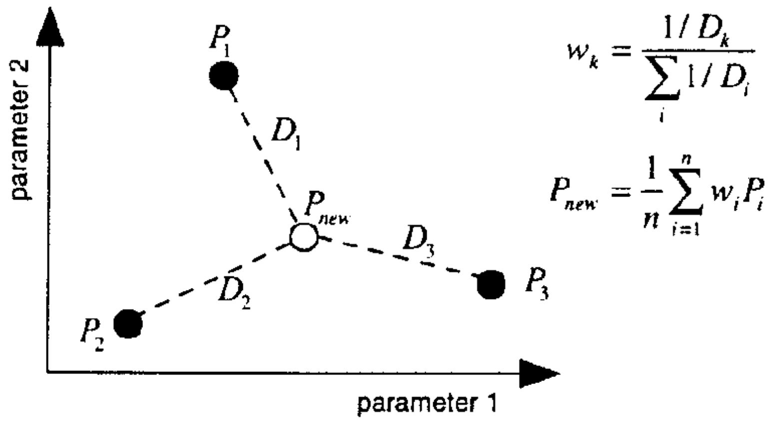


Figure 47: Ohsaki's method for subjective performance prediction using K-Nearest Neighbors [117]

The K Nearest Neighbor fitness prediction technique with Euclidean distances works well for problems with continuous input variables, but cannot be applied to problems with discrete or hierarchical encodings for two reasons. The first issue is that for problems with structured encodings the distance between inactive genes is insignificant, and their inclusion would add excessive noise to the problem. Secondly, the Euclidean distance cannot be used to measure similarity between discrete and categorical variables such as engine cycle types or structural concepts.

In response to this difficulty, a new measure of solution similarity D_s was formulated that takes into account the nature of a structured genetic representation. Calculation of the structured distance D_s between two solutions involves two steps. First, the number of discrete genes that do not match are recorded as H, the Hamming distance. (Equation 11)

$$H = |\bar{X} \backslash \bar{Y}| \tag{11}$$

where $\bar{X}$ and $\bar{Y}$ are the vectors of categorical variables associated with each respective parent. The second step involves the calculation of D_e, the normalized Euclidean distance between the genes common to both continuous sets. (Equation 12)

$$D_e = \frac{\sqrt{\sum (\bar{x} - \bar{y})^2}}{|\bar{x}|} \tag{12}$$

where $\bar{x}$ and $\bar{y}$ are the vectors of continuous design variables associated with each parent. The value of D_s is then found by adding the values of H and D_e.

The effectiveness of this new measure of solution similarity was tested in comparison to the traditional Euclidean distance measure for an aircraft test problem with a structured encoding. A total of four hundred designs were presented to the author for subjective evaluation using the GUI shown in Figure 28. Three hundred of these points were used as the training set for a K-nearest neighbor classifier with both the traditional Euclidean distance measure and the new measure of solution similarity D_s. The remaining one hundred points were used to test the effectiveness of both classification systems for both similarity measures. The results, plotted in Figure 48, show that the predictor that uses D_s significantly out-performs the traditional Euclidean distance for hierarchical classification problems. Additionally, the results indicate that a value of K between 15-30 yields the best performance for a training set with 300 data points.

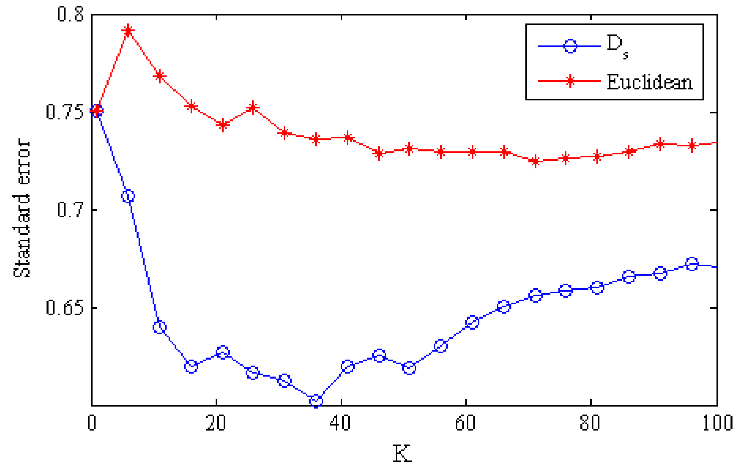


Figure 48: Impact of different similarity measures and values of K on K-nearest neighbor classification accuracy

4.1.2 Method description

The new Hybrid Interactive Genetic Algorithm created during the course of this research builds upon the SPEA2 algorithm discussed in Section 2.4.2.2, Interactive Genetic Algorithms, and the Injection Island GA, resulting in a system that blends expert preference with numerical optimization. (Figure 49) The algorithm consists of two main islands: one which is interactive, and a second which uses KNN interpolation as a surrogate for the expert. After the interactive population has been run for a user specified number of generations, a KNN model is created using the designs that have been evaluated by the human so far. This model is then used by the secondary population, which can be run for a much larger number of generations because no human intervention is required. Once the secondary population's termination criteria has been met, its best designs are presented to the decision maker and then injected into the primary population. After injection, the primary interactive algorithm proceeds as normal until either the secondary population initialization criteria its own termination criteria is met.

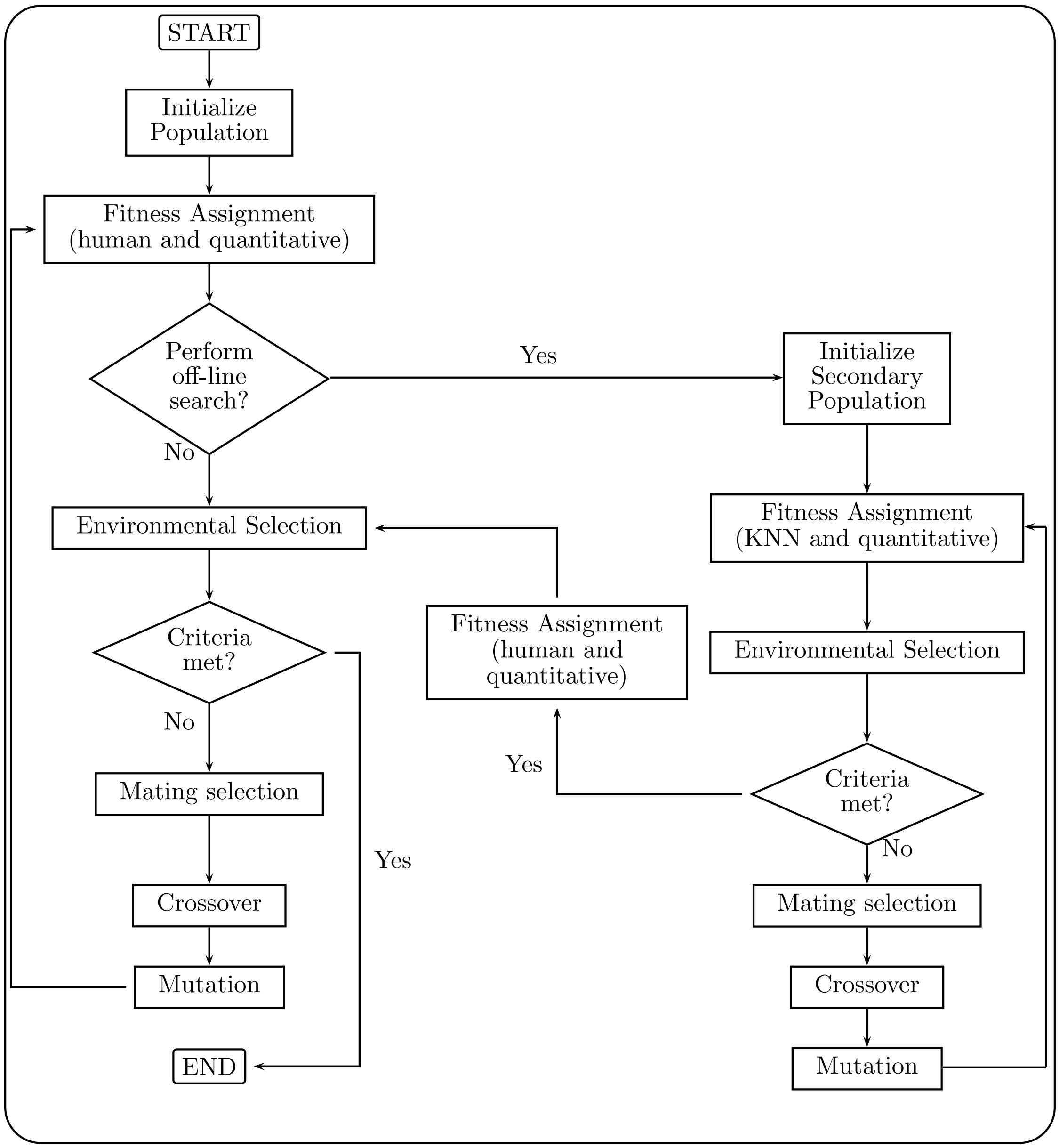


Figure 49: The Hybrid Interactive Genetic Algorithm

4.1.3 Method demonstration

In order to demonstrate the ability of the Hybrid Interactive Genetic Algorithm to solve problems with qualitative and quantitative criteria, the method has been applied to a simplified sonic boom optimization problem that has been previously demonstrated to lead to infeasible results.

In [97], Bandte used a broad IEC system to optimize response surface equations of a supersonic business jet that were developed in [20]. Although it was demonstrated that Genetic Algorithms could be used to find designs with low sonic boom levels, the wing planform shape selected by the algorithm, shown in Figure 43(b), is obviously infeasible. This unusable design is a consequence of the fact that no stability constraints were present in the model.

The demonstrated deficiency of non-interactive methods for this problem along with the fact that the same response surface equations are available to the author make this optimization task an ideal demonstration case for the new Hybrid Interactive Genetic Algorithm. To simply the problem, it is assumed that only those variables directly related to planform geometry influence the decision-maker's preference for a particular concept. As a result, only a planform view of the wing need be presented to the decision-maker. This presentation was accomplished using the GUI depicted in Figure 50. The demonstration problem is formulated as given in Table 4:

The algorithm was coded in MATLAB, and run using the parameter settings listed in Table 5. Total execution time, including human fitness evaluation, was approximately 45 minutes. Figure 51 presents the Pareto optimal designs present

Figure 50: GUI used to obtain user ratings for the Hybrid Interactive Genetic Algorithm demonstration problem

Table 4: HIGA demonstration problem definition

Maximize:	user satisfaction
Minimize:	sonic boom loudness
By varying:	decision variables (Table 6)
Subject to:	side constraints (Table 6)

in the primary population's final generation, along with all of the designs present in the 1^{st} generation. From the figure, it is clear that the algorithm was able to successfully locate a tradeoff surface between sonic boom level and user satisfaction with the concept. Interestingly, upon inspection of the results three of the final Pareto optimal concepts appear to be very similar to each other, yet have subjective fitness scores that range from 3 to 5. This is an indication of the stochastic nature of the human preference rating, and may be due to time variant or psychological effects. For example, if one had been recently asked to assign fitness to a number of very "desirable" concepts, one might be inclined to discount another design which would otherwise be deemed "acceptable". Although these psychological aspects of human fitness assignment are interesting and merit further study, the results of this demonstration problem demonstrate that the Hybrid Interactive Genetic Algorithm can still be effective for problems with both subjective and quantitative criteria.

Table 5: Parameter settings for the HIGA demonstration problem

	Primary population	Secondary population
Population size	20	100
Crossover rate	100%	70%
Crossover operator	BLX-0.5	BLX-0.5
Mutation rate	2%	2%
Mutation type	uniform	uniform
Termination criteria	400 function evaluations	50 generations

Table 6: Design variables and ranges used for the Hybrid Interactive Genetic Algorithm demonstration problem

Variable name	Lower Bound	Upper Bound
Engine Location (ft)	100	110
Wing Location (ft)	45	57
Cabin Location (ft)	36	41
Empennage Location (ft)	87	97
Cabin Length (ft)	39	50
Fuselage Length (ft)	135	160
Aspect Ratio	2	2.5
Taper Ratio	0.05	0.3
Planform Area (ft^2)	2300	3100
Wing Sweep	67	74
Strake-Body Intersection	0.4	0.8
Strake-Wing Intersection	0.2	0.4
Aft Strake- Body Int.	0.4	0.6
Aft Strake-Wing Int.	0.2	0.5
Root Thickness/Chord	0.025	0.045
Tip Thickness/Chord	0.025	0.035
Root Twist (^{o})	-2	2
Tip Twist (^{o})	0	5
Diameter 1 (ft)	2.2	3
Diameter 2 (ft)	7.2	7.6
Diameter 3 (ft)	7.2	8
Diameter 4 (ft)	7.2	7.6
Diameter 5 (ft)	4.5	6.5
Diameter 6 (ft)	2.3	3.1
Overall Pressure Ratio	22	29
Turbine Inlet Temp (^{o}R)	3300	3400
Fan Pressure Ratio	2.6	3.2
Throttle Ratio	1.2	1.23
Aircraft Thrust/Weight ratio	0.41	0.45
Number of Passengers	8	12
Manuf. Return on Investment	6	12
Number of Vehicles Produced	200	500
Design Range (nm)	3500	4200
Design Mach Number	1.6	1.8
Takeoff Thrust Derating	0.8	1

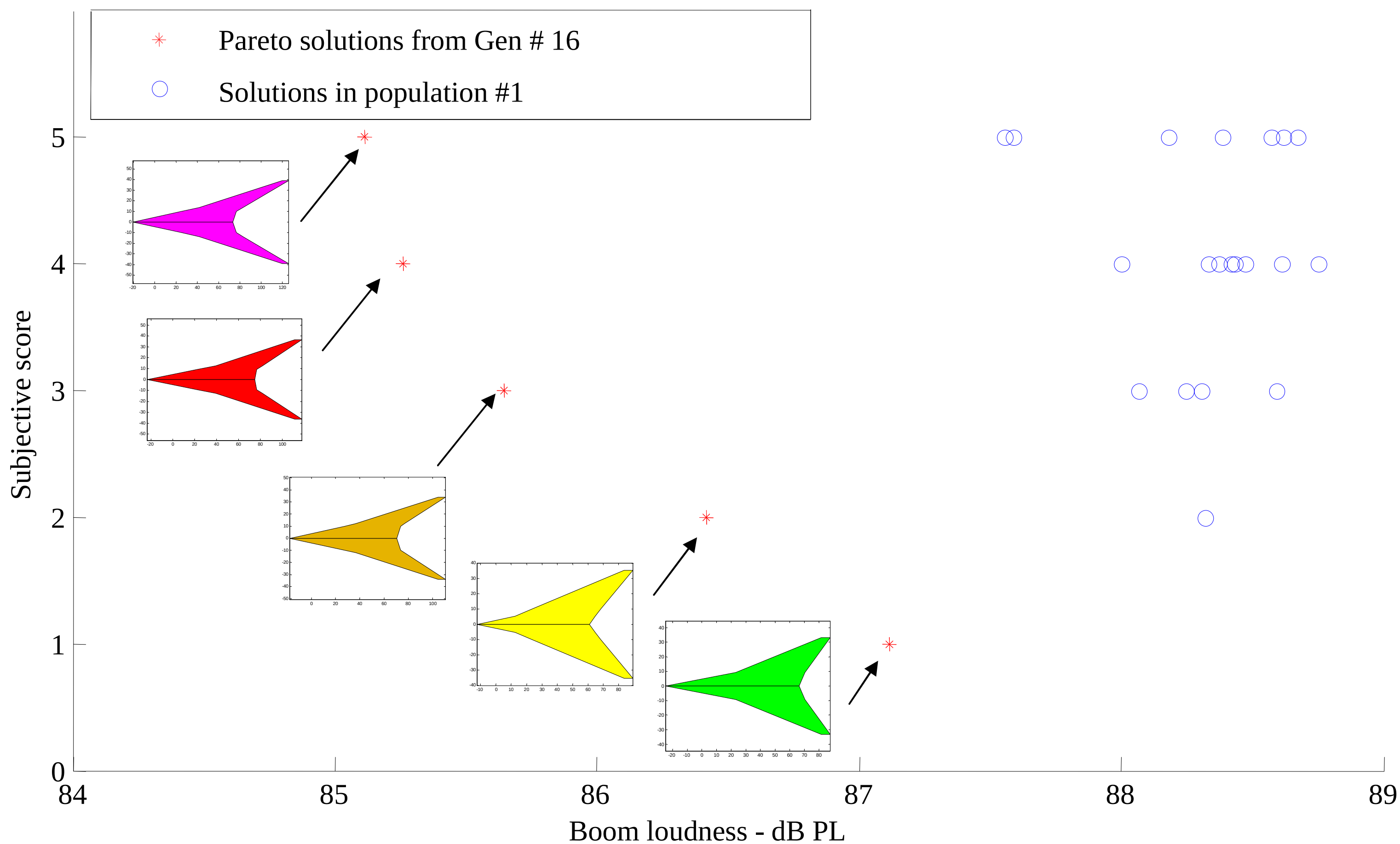


Figure 51: Pareto tradeoff surface obtained using the Hybrid Interactive Genetic Algorithm

4.2 Hierarchical System Design using Evolutionary Algorithms (Question 2)

> *Is it possible to rigorously search the Matrix of Alternatives for promising concepts without individually optimizing every possible alternative?*

Although there have been hundreds of applications of Evolutionary Algorithms to engineering design problems, very few researchers have tackled whole system design problems in which both the system architecture and design variables are optimized concurrently.

One of the only applications found in literature is the application of the structured Genetic Algorithm to the design of a hydropower plant. [119] Unfortunately, it was found that the Structured GA quickly focused on a single alternative, even for relatively small design hierarchies. The tendency to quickly focus on a small portion of the design space may result in the algorithm overlooking potentially promising solutions. This deficiency was addressed with a method called the hybrid structured Genetic Algorithm, which introduced variable mutation probabilities to the morphology and decision variables. [119] This method used a bitwise mutation probability of 20% for morphological variables such as dam site or type, and a much lower mutation probability of 2% for continuous decision variables such as tunnel lengths. The large mutation probability applied to the morphological variables effectively maintained genetic diversity, but may also prevent efficient convergence because of the mutation operator's propensity to destroy good solutions. Another difficulty with this approach is that crossover between designs of very different discrete configurations is not likely to improve the values of the continuous variables, leading the algorithm to act like a simple hill climber. Finally, the continuous version of the Structured Genetic Algorithm used in [119] does not respect the guidelines for continuous crossover operators established by [37]:

1. Population's mean decision variable vector should remain the same before and after recombination

2. Variance of the inter-member distances must increase after recombination

4.2.1 Method formulation

The proposed method of tackling hierarchical synthesis problems associated with this work retains the structured Genetic Algorithm as a foundation, but does not rely on large mutation probabilities to maintain genetic diversity. Instead, the method uses modified methods of mating selection and replacement in addition to a new crossover operator that improves the efficiency of recombination for hierarchical problems.

Hierarchical Crossover Operator: Given two solutions with structured encodings, two children solutions are produced using the hierarchical crossover operator described in Algorithm 1. First, the the algorithm checks for similarity among the discrete genes of the two parent solutions. In the case where both solutions share the same discrete value, the continuous subset of the alternative common to both solutions is recombined via the BLX-α operator described in Section 2.3.5.1. If the parent concepts do not share a common value, a random number is drawn from [0 1]. If the value is greater than 0.5, then nothing happens and the algorithm proceeds to the next discrete gene. In the case where the random number drawn is less than 0.5, the discrete values and their associated continuous sets are swapped.

Algorithm 1 Perform structured crossover

Input: Parent solutions $x^{(1,t)}$ and $x^{(2,t)}$
for $i = 1$ to $n_{discrete}$ **do**
  if $x_i^{(1,t)} = x_i^{(2,t)}$ (Two solutions share a common control gene) **then**
    $x_{i,1:n}^{(1,t+1)}, x_{i,1:n}^{(2,t+1)} = BLX_\alpha(x_{i,1:n}^{(1,t)}, x_{i,1:n}^{(2,t)})$
  else
    if $rand < .5$ (50 % of the time) **then**
      Swap the control genes:
      $x_i^{(1,t+1)} = x_i^{(2,t)}$
      $x_i^{(2,t+1)} = x_i^{(1,t)}$
      Swap the associated continuous genes:
      $x_{i,1:n}^{(1,t+1)} = x_{i,1:n}^{(2,t)}$
      $x_{i,1:n}^{(2,t+1)} = x_{i,1:n}^{(1,t)}$
    end if
  end if
end for

Both the swap and the BLX-α operators used in the new structured crossover algorithm respect Deb's first suggestion that the mean of the parent and children decision vectors should be the same. Additionally, the use of an α value greater than 0 with the BLX-α operator ensures that the variance of children solution's decision variables will be greater than that of their parents.

Diversity preservation and mating selection: The introduction of a new crossover operator is not by itself sufficient to solve the difficulties that have been encountered in the course of previous GA concept selection applications. Specifically, the structured crossover operator does not prevent the algorithm from quickly focusing on a very small subset of the matrix of alternatives, possibly resulting in an incomplete search of the alternatives space. In [119], this was addressed through the introduction of very large mutation probabilities, but this has been shown to be disruptive and leads to greater computational expense.

The tendency of GAs to converge prematurely is not unique to hierarchical applications, and several researchers have investigated alternative methods to maintain population diversity. In [36], DeJong proposed a method known as crowding to preserve genetic diversity. Unlike the standard GA in which children solutions automatically replace their parent solutions, a GA with crowding compares the genotype of each child solution with CF members of the parent population and replaces the population member which is most similar to the child solution. For small values of CF, a GA with crowding will act in a similar fashion to the standard GA, but when large values are used, crowding effectively maintains population diversity. A modification of DeJong's crowding method called Restricted Tournament Replacement (RTR) has been used to perform a similar function in the present method. [62] Under the RTR method, each child solution is compared with ω solutions from the parent population, and replaces the closest of these if it has better fitness.

Although the introduction of a crowding mechanism helps to maintain a diverse population and prevent premature convergence, it does not necessarily improve the performance of the algorithm. This is because the diverse nature of the population will prevent efficient evolution of the continuous parameter set, as it is unlikely that two designs of similar hierarchies will be selected for crossover. For example, examination of Algorithm 1 reveals that when two parent solutions share no common control genes, branches of their hierarchies may be swapped but no change will be made to any design variables.

In order to remedy this difficulty, a mating restriction scheme developed by

Ishibuchi and Shibata [77] is adopted for use in the present method. This mating selection algorithm biases the selection phase of the algorithm to prefer and cross more phenotypically similar parents. Under this mating restriction scheme, shown in Figure 52, α solutions are chosen via standard binary tournament selection with replacement. Out of these α candidates, the solution which is most similar or dissimilar from the others, depending on the sign of α, is chosen as the first parent. The mate for this first parent is biased by performing β fitness-based tournament selections and then selecting the winner closest to (or furthest from) from the original parent, as measured in genotypic (decision) space. By increasing the magnitude of β, the user increases the strength of similarity or dissimilarity between parent solutions. Positive values of β are of special interest for problems with hierarchical encodings because crossover between similar solutions is expected to produce children with better fitness values.

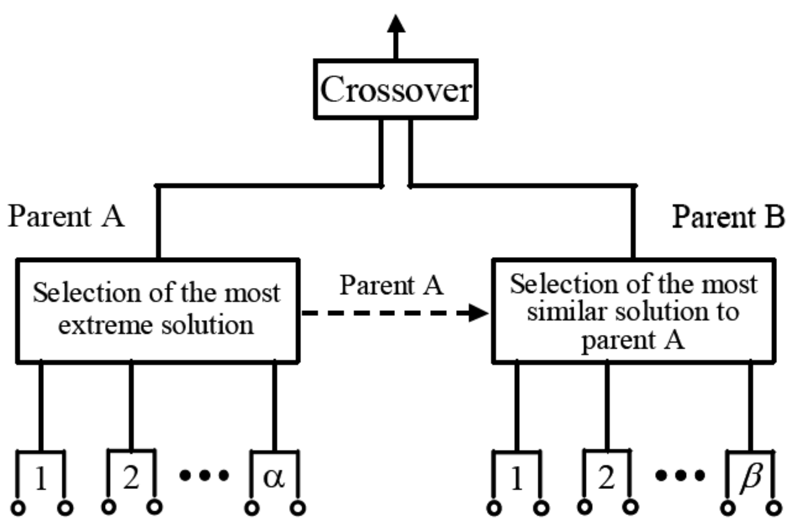


Figure 52: Tournament Selection procedure with mating restriction [77]

The modified structured Genetic Algorithm, shown in Figure 53, has been designed to preserve population diversity and prevent premature convergence while still allowing for efficient optimization of the continuous parameter set. In contrast with the hybrid structured Genetic Algorithm, the new method makes use of Restricted Tournament Replacement rather than high mutation probabilities to prevent premature convergence. Additionally, the use of biased mating selection and a new hierarchical crossover operator minimize the number of "lethals", or unfit individuals, that are the product of recombination.

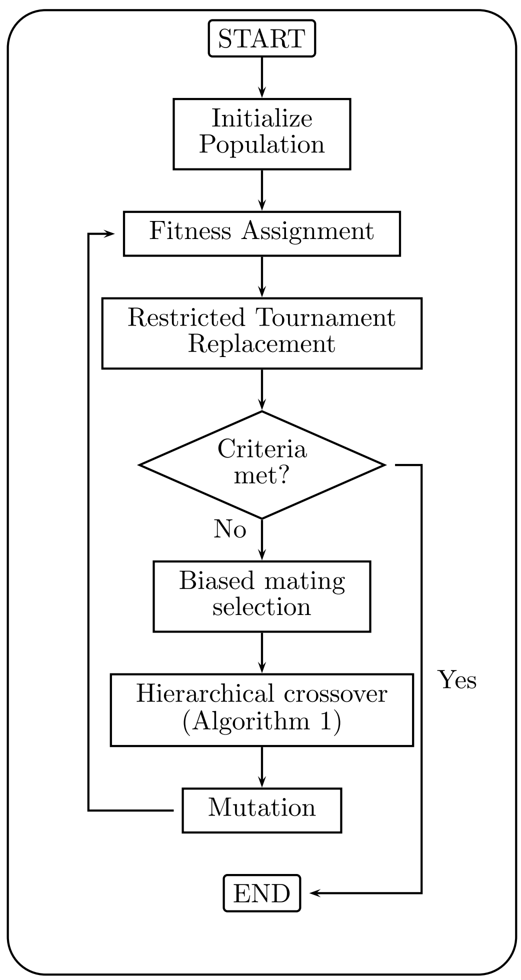


Figure 53: The modified Structured Genetic Algorithm

4.2.2 Method demonstration

In order to demonstrate the utility of the modified structured Genetic Algorithm, the method was compared to the performance of the hybrid structured Genetic Algorithm as described in [119]. One issue that complicates the task of comparing the performance of algorithms designed to solve problems with structured encodings is the lack of a computationally inexpensive test problems. Most papers in the literature use a suite of very simple closed-form equations to test the performance of new algorithms, but these functions cannot be used to test the efficiency of the present method.

Because a search of the literature revealed no suitable test problems, an aircraft optimization problem using the physics-based analysis tools described in Section 5.3 was used to compare the performance of the modified algorithm with that of the the original structured GA. Ten different objectives were aggregated using an l_p metric, and optimized using both the original and modified sGA using the parameter settings listed in Table 7. The modified algorithm was evaluated using two different combinations of settings for the mating selection control parameters α and β.

The design hierarchy searched by the algorithms consists of six categorical variables, combinations of which result in a possible 576 system architectures as displayed in Table 12. These different alternatives are associated with a number of continuous design variables listed in Table 13.

The goal of the design hierarchy search is the identification of high performance concepts and not necessarily detailed parameter optimization of each design variable.

Table 7: Parameter settings for the msGA demonstration problem

	msGA (run 1)	msGA (run 2)	hsGA
Population size		200	
Crossover rate		70 %	
Crossover operator	BLX-0.5	BLX-0.5	uniform
Mutation rate (continuous)		2%	
Mutation rate (discrete)	2%	2%	20%
Mutation type		uniform	
Mating similarity parameter α	1	-3	-
Mating similarity parameter β	3	5	-
RTR window size ω	200	200	-
Termination criteria		100 generations	

However, meaningful comparison necessitates that these parameters are at least somewhat optimized or incorrect conclusions may be drawn, as mentioned in Section 2.2.1. Because of this phenomenon, algorithms used to search the design hierarchies must be efficient at both feature selection and parameter optimization.

Figure 54 presents the test problem results of each of the three algorithms. In the figure, each data point represents one of the 576 possible vehicle concepts as defined in Table 12. In the figure, the horizontal axis corresponds with the best fitness value assigned to a particular branch of the concept tree, and the vertical axis gives the number of function calls allocated to that particular branch. A measure of algorithm efficiency could be considered to be simultaneous minimization of fitness and function calls. It is evident from Figure 54 that the modified algorithm with α=1 and β=3 is more efficient than the other two tested. Although the negative value of α used for the second trial of the present method was expected to improve the thoroughness of the search, it apparently has done so at the expense of efficiency. However, the fact that a different β value was also used makes this result inconclusive.

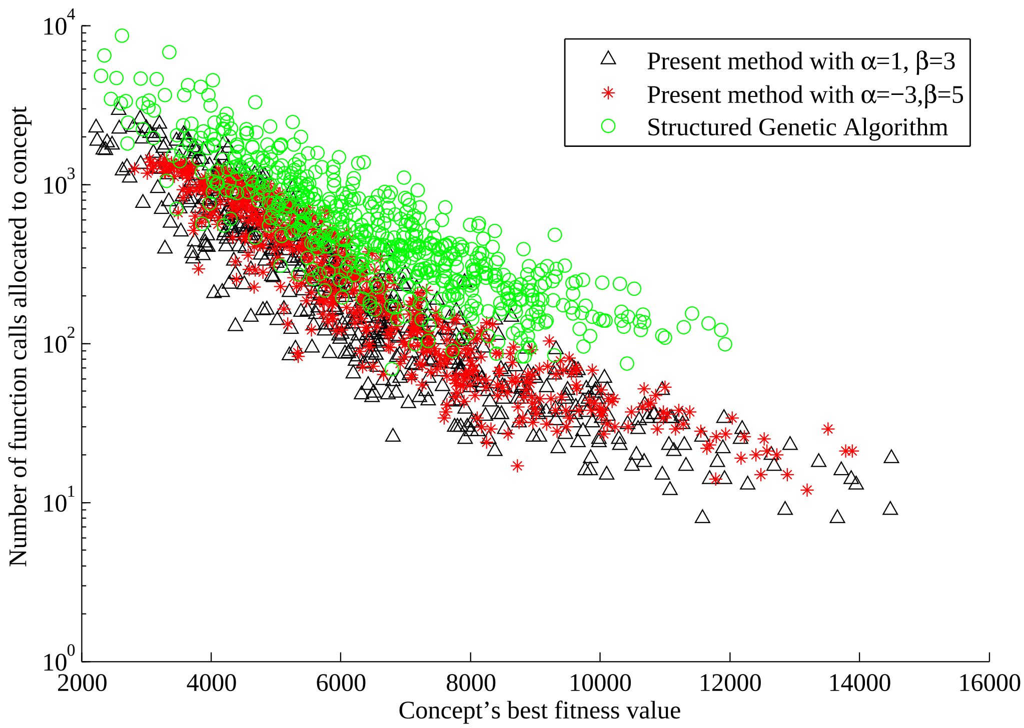


Figure 54: Comparison of the efficiency of the modified and hybrid structured Genetic Algorithms

4.3 Accelerating Multiobjective Evolutionary Algorithm Convergence (Question 3)

Is there a computationally feasible way to use physics-based analysis tools in addition to the historically-based design guidelines commonly used during the concept generation phase of routine design programs?

The most prominent criticism of evolutionary optimization is that these methods typically require more function evaluations than classical optimization techniques. This limitation is even more severe for multiobjective evolutionary algorithms because of the fact that exponential population size increases are required as the number of objectives are increased. [82]

4.3.1 Incorporation of preference information

One of the reasons why multiobjective algorithms have difficulties solving problems with more than two or three criteria is the reliance upon dominance-based fitness assignment. According to the definition of non-dominance, a solution is non-dominated if no other solution in the set is better in any objective without being inferior in at least one other objective. Because the definition of dominance does not deal with magnitudes, a solution which trades a near infinite amount of capability in one objective for an infinitesimal amount of gain in another would not be penalized by the fitness assignment schemes used by many of the common MOEAs in literature. For problems with a small number of objectives this will likely not be a significant problem because a large portion of the final population will typically lie in a region of "significant tradeoffs". However, as the number of objectives are increased, this fraction of the population that lies in an "interesting" region will decay exponentially. The result is that most of the final designs presented to the decision maker will be uninteresting because they exhibit unacceptable performance in at least one metric. Several methods of incorporating preferences into evolutionary algorithms have been discussed in literature (e.g. [18] [30]), but they are not well suited for problems with many objectives because the amount of preference information required by the decision-maker scales poorly with the number of objectives.

4.3.1.1 Multi-value Genetic Algorithm

In response to the poor performance the MOEAs available in literature exhibit for problems with many objectives, the author developed a method called the Multi-utility Genetic Algorithm. The principle behind the algorithm is quite simple: rather than optimizing each of the M objectives, the problem is re-formulated to solve for the tradeoff between M biased aggregate functions that favor attainment of one particular objective but do not ignore performance in the other $M-1$ objectives. Although any aggregation technique such as the weighted sum or l_p metric can be used, the present work has relied upon Goal Programming (see Section 2.4.1.3).

In order to proceed with the method, the decision-maker must initially only specify two parameters per objective: a goal, or ideal value, and a threshold, or worst practical value. The goal value is defined as the level of performance. The threshold value corresponds to the minimum level of performance in a given metric that would be considered acceptable by the decision-maker. In the case where the M different objectives have different variances, a vector of normalization constants are also necessary, but can be obtained via sampling. Using these vectors of goal and threshold values, M value functions are formulated as Equation 13:

$$V_i(x) = \left(\sum\nolimits_{m=1}^{M} w_m \left| \frac{f_m(x) - t_m}{n_m} - \frac{f_i(x) - t_i}{n_m} + \frac{f_i(x) - g_i}{\frac{n_m}{\alpha}} \right|^p \right)^{1/p} \tag{13}$$

where $\bar{w}$ is a vector of weights, $\bar{t}$ is a vector of threshold values, $\bar{g}$ is a vector of goal values, $\bar{n}$ is a vector of normalization constants, and α is a parameter that controls the relative contribution of goal achievement to the value function.

4.3.2 Parallelization of MOEAs

One obvious way to increase MOEA performance is to parallelize the algorithm by using multiple processors. During the discussion of parallel EA's in Section 2.3.5.2, three broad categories of parallelization schemes were introduced: master/slave, coarse grained (island), and fine grained (diffusion).

The coarse-grained GA model is one of the most popular methods for parallelization of single-objective evolutionary algorithms, but its direct application to multiple objective problems can be inefficient because of multiple objective population size scaling issues. For single objective problems, coarse-grained EAs are typically run using fewer individuals on each island because the independent nature of the islands maintains sufficient diversity without excessive individual population sizes. Unfortunately, this strategy will result in low selection pressure for problems with a large number of objectives because all solutions will likely be non-dominated with respect to each other. Fine grained parallel MOEA models are likely to encounter similar difficulties because solutions are only compared to a small number of individuals in close proximity within the fine-grained population structure.

As with single objective MOEAs, the master/slave method of parallelization is the most simple and intuitive method of adding parallel capability to a MOEA. Although the master/slave method may have more communication overhead than coarse-grained parallelization methods, this issue is typically insignificant for real world engineering problems because the CPU time required to complete the numerical analysis far outweighs the delays associated with networking. An additional benefit of the master/slave model is that it can be very flexible, especially in the case where a very large number of CPUs are available for the analysis.

4.3.3 Method Demonstration

The ability of multi-value genetic algorithms to better solve problems with large numbers of responses has been demonstrated through comparison with the popular SPEA2 algorithm. The test problem for the comparison has five objectives, and is represented using response surface equations. To ensure as fair a comparison as possible, both of the algorithms tested used the same parameter settings where applicable as listed in Table 8. The MVGA also required the specification of goal and threshold performance settings as described in Section 4.3.1.1, and these values are given in Table 9.

The results of both algorithms are presented in scatterplot form as Figure 55. It is evident from the figure that the MVGA algorithm's solutions consistently dominate those produced by the SPEA2 algorithm.

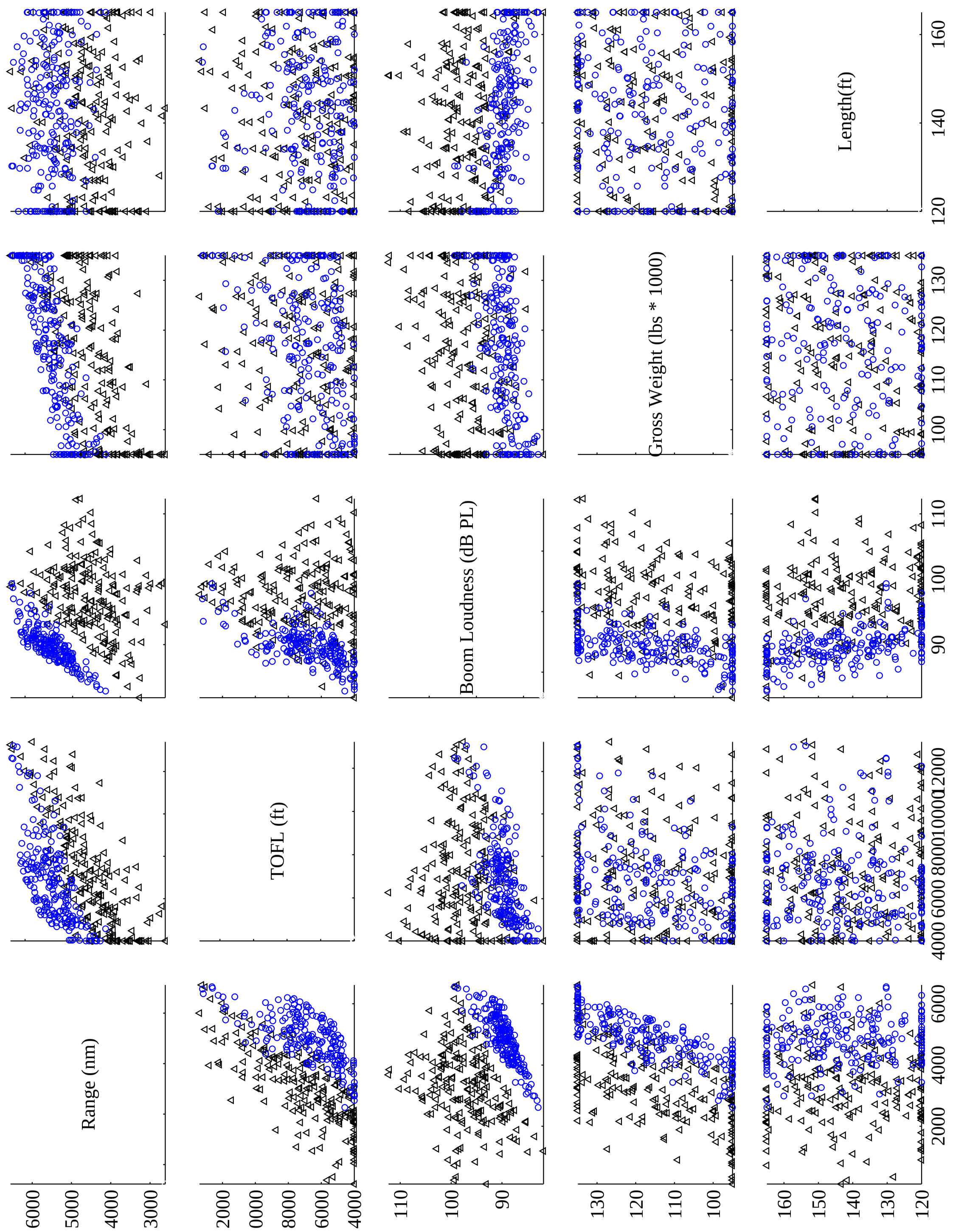


Figure 55: Scatter-plot of Pareto hypersurfaces generated by the SPEA2 algorithm (black triangle) and the present method (blue circle)

Table 8: Parameter settings for the MVGA demonstration problem

	MVGA	SPEA2
Population size	200	200
Crossover rate	70%	70%
Crossover operator	BLX-0.5	BLX-0.5
Mutation rate (continuous)	2%	2%
Mutation type	uniform	uniform
Termination criteria	50,000 function calls	50,000 function calls

Table 9: Goal and threshold levels used by the MVGA value functions

Objective	Goal	Threshold
Range (nm)	6500	4000
TOFL (ft)	6000	8000
Boom Loudness (dB PL)	82	88
Gross Weight (lb)	100,000	125,000
Length (ft)	130	140

4.4 Concept Exploration using Multi-objective Interactive Genetic Algorithms

A method to assist in the task of concept exploration has been created by combining the capabilities developed in the preceding sections . This method, shown in Figure 57, shares many attributes with current design methods, but features several innovative techniques designed to provide the engineer with insight into the relationship between requirements and attributes for creative design problems.

Multi-value optimization: The fifth step of the method pictured in Figure 57 is the first point at which the proposed concept exploration method differs significantly from more traditional design efforts. Although goals for each objective are established in step 2 of the method, these values are not treated as hard constraints because of the potential to over-constrain the problem. By using a multi-objective approach, the designer can explore the relationship between objectives and make more informed decisions regarding appropriate requirements.

Multi-objective Interactive Optimization: In the case where the multi-value optimization step does not yield a solution that is both feasible and acceptable, step 7 can be used to further explore the concept space in search of a design that meets these criteria. By using the results of the Pareto hypersurface investigation performed in step 6, the design goals defined in step 2 can be refined and used to form a single realistic value function. This value function is used as one objective in a bi-criteria problem which solves for the tradeoff between numerical performance as defined by the value function and acceptability as defined by the analyst. The generation of acceptable solutions is assured by allowing the designer to provide feedback to the algorithm during the course of the search.

The concept exploration method's domain of applicability largely corresponds with Phase 0 of the DoD acquisition process as shown in Figure 56. Although elements of the method may be useful in other stages of the design process, its capabilities are best suited to use with the computationally inexpensive analysis commonly used during this stage of a design program.

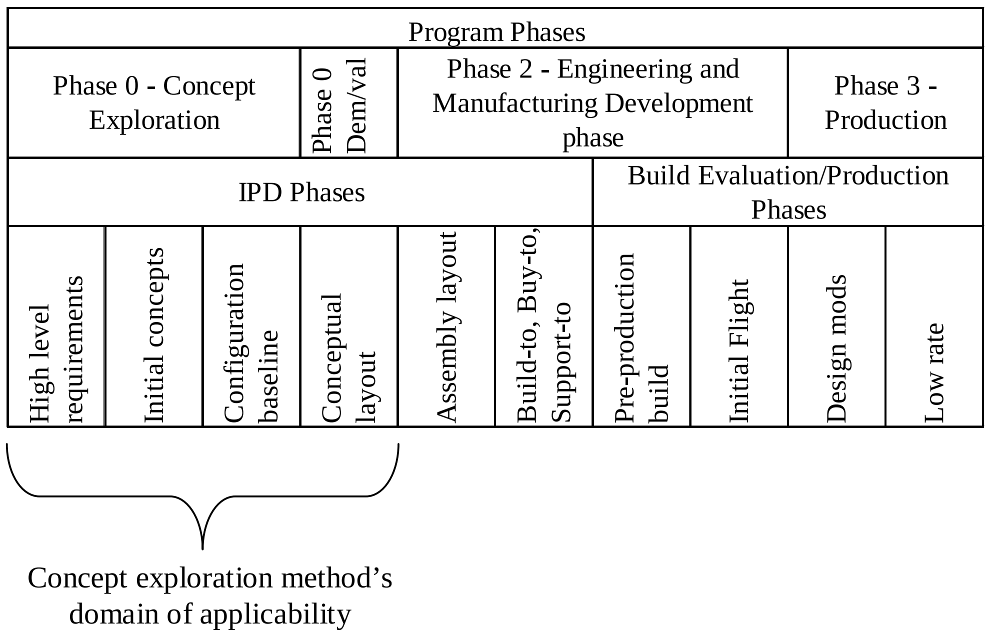


Figure 56: Illustration of the concept exploration method's domain of applicability (Modified from [3])

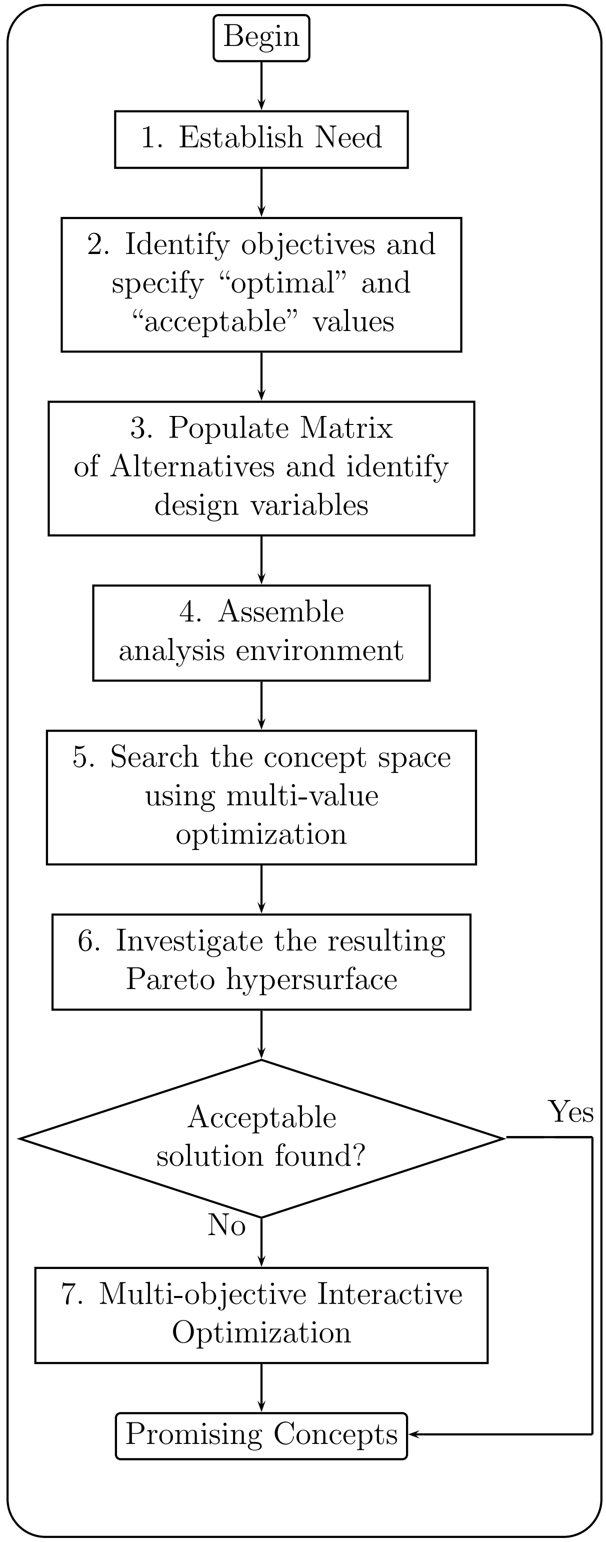


Figure 57: Process flow of the proposed concept exploration method

CHAPTER V

METHOD IMPLEMENTATION

Most of the major airframe manufacturers, spurred on by the favorable market forecasts and the DARPA QSP program [160], have researched quiet supersonic airplanes, but to this point no definitive conclusion has been reached regarding what the vehicle should look like. Designs proposed by industry members and government differ in engine location, wing planform type and location, control surface layout, engine cycle, and other key defining characteristics. A survey of these efforts reveals that in addition to physical and geometric differences, each organization has quite different goals in mind, as shown in Figure 58. These differing requirements are likely a driving force behind the widely varying configurations currently being proposed.

The lack of expert consensus about what a small, quiet supersonic airplane's requirements and configuration should be make it an ideal demonstration problem for the method under development, because no historical guidance is available to provide insight into the impact of these important early decisions on project goals.

5.1 Challenges associated with supersonic business jet design

Several market research studies have concluded that there is a large demand for a supersonic business jet given that the significant technical challenges associated with such a vehicle can be overcome.

Requirement	Aerion	Gulfstream	NASA	Raytheon	SAI
Range (nm)	4000	4800	4000-5000	5000	4000
Gross Weight(lbs)	90,000	100,000	100,000	120,000	150,000
Balanced Field Length (ft)	6000	6500	6500	6000	8000
Cruise Mach Number	.99/1.6	1.6-2.0	1.6-2.0	1.8	1.6-1.8
Payload	8-10 pax	1500 lbs	6-10 pax	6 pax	8-12 pax
Length (ft)	145	140	130-140	165	130
Takeoff/Landing Noise	Stage IV	Stage IV-10dB	Stage IV-2dB	Stage IV	Stage IV
Sonic Boom	no constraint	.15 psf ramp	.4 psf ramp	.4 psf ramp	.3 psf ramp

Figure 58: Recent Small supersonic transport design requirements published by industry and government

5.1.1 Quiet supersonic overland capability

A number of independent market assessments have concluded that the business case for the small supersonic transport is dependent upon the ability to fly supersonic overland. [65][107][151] This requirement is perhaps the most significant obstacle in the path of the vehicle's development because supersonic overland flight is currently prohibited under 14 CFR Part 91. Apart from these regulatory and legal issues, the technical challenges of designing a low-boom aircraft that simultaneously performs well in other aspects are substantial.

The fact that an aircraft's sonic boom is proportional to its weight divided by the three-halves power of its length means that supersonic business jets will naturally have lower boom levels than larger vehicles such as the Concorde. The reduction in weight alone, however, will not yield boom levels low enough to be considered acceptable. Fortunately, a sonic boom minimization theory developed by Seabass and George [136] and Darden [31] can be applied to solve for an equivalent area distribution resulting in minimum overpressure or shock pressure rise. By carefully tailoring the vehicle's volume and lift distribution to match this target, low sonic boom levels can be achieved. The validity of this theory has been experimentally validated both in wind tunnels and in flight tests, but whether or not this theory will be sufficient to design supersonic vehicle's with sufficiently quiet booms is not yet known. As noted in [95]:

> While there is a minimum sonic boom theory for the shaping of the configurations forward 75 to 80 percent ... there are no other theoretically

and experimentally- validated methods for controlling and reducing the tail shock in a manner similar to that of the nose shock.

Another issue complicating the design of a low boom supersonic transport is the fact that no concrete definition of an "acceptable" boom has been established. The DARPA QSP program suggested that a .3 psf shock pressure rise, but several researchers have indicated that shock pressure rise in fact correlates poorly with subjective response to sonic booms. [90] Human tests using real and simulated booms have found that metrics based upon Perceived Loudness (PL) or A-weighted sound pressure level dBA may be the best indicators [145] and others have suggested that a loudness equivalent to 84-88 dB PL or 68 dBA may be acceptable for overland supersonic flight. [90] The FAA is in the process of amending Part 91 and determining what if any sonic boom level would be acceptable for overland supersonic flight, but until these guidelines are published engineers are designing towards an unknown target. [51]

5.1.2 Accessibility - range, field performance, community noise, gross weight, and length

In a broad sense, the term accessability can be used to encompass many of the performance metrics critical to market acceptance of a small supersonic transport. Ultimately, the customer cares about whether or not a given vehicle will be able to transport them from point A to point B, and factors such as range, field performance, community noise, and the vehicle's gross weight all factor into this criterion. Unfortunately, all of these objectives traditionally conflict with each other, and several of the issues associated with supersonic vehicles in particular only exacerbate the problem.

Range: Market research has indicated that in order to be successful, a supersonic business jet must be capable of travelling a minimum of 4000 nautical miles with IFR reserves, and a range of up to 5000 nm is highly desirable. [161] Although achieving this long range capability is by itself not technically challenging, the long range requirement will tend to negatively impact the other evaluation criteria. For example, long range supersonic aircraft also tend to have low aspect ratio wings and high wing loadings, resulting in poor field performance. The requirement for long range also leads to the necessity for low empty weight fractions and consequently high vehicle weight.

Gross Weight: Minimization of vehicle gross weight has often been chosen as the objective in aircraft design exercises because nearly all criteria other than payload-range benefit from weight reduction. Gross weight can also have an impact on access-ability because heavy aircraft may be prohibited from landing at some airports due to local regulations or pavement loading restrictions. Examination of the airport weight and field length limitations shown in Figure 59 reveals that gross weight limitations may in fact prevent heavy vehicles from operating out of some of the most popular GA airports. In particular, the 100,000 lb weight limit at Teterboro and several other popular GA destinations has been cited as a significant constraining factor to supersonic business jet design.

Vehicle length: Although perhaps not a traditional metric, vehicle length will be an important consideration during the design of a low-boom supersonic business jet. Many of the general aviation airports may be poorly equipped to handle a vehicle

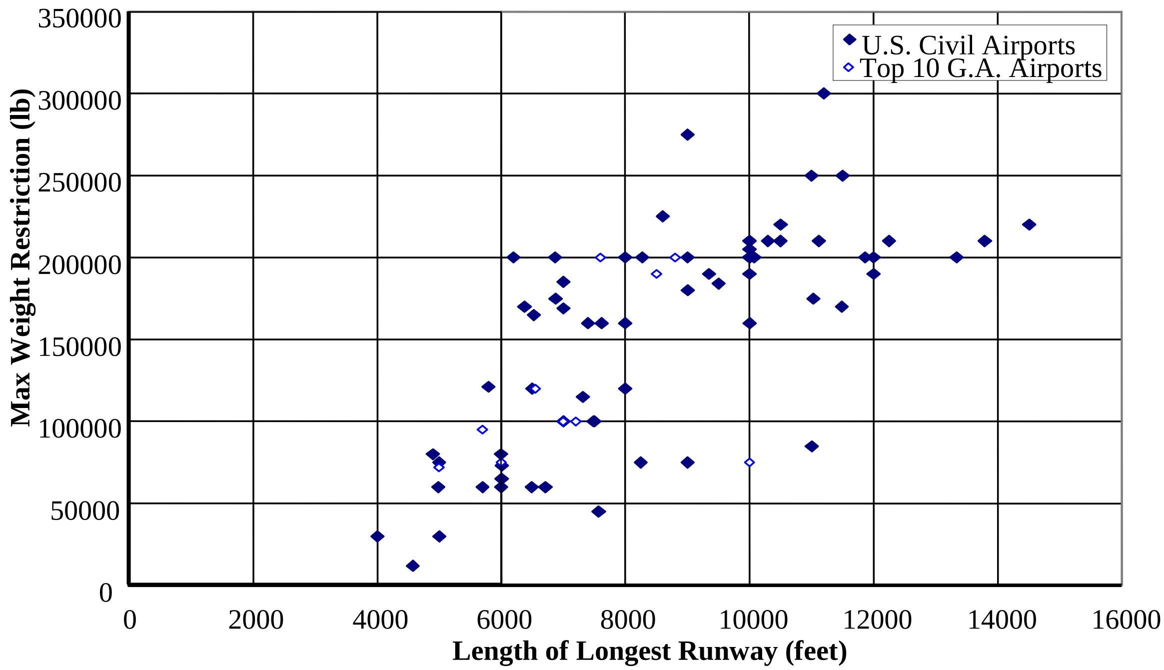


Figure 59: Joint distribution of runway lengths and gross weight limitations for popular GA airports(Adapted from [128])

that will be up to 65 ft longer than current subsonic business jets. In [131] it is noted that vehicle weight has historically varied with the square of length. This trend will likely not be as severe for supersonic business jets, but high fineness-ratio fuselages will still be associated with an weight penalty, negating at least a portion of the sonic boom benefit associated with long vehicles.

Field Performance: The tradeoff between field performance and cruise performance is perhaps one of the best understood of all compromises required of an aircraft designer. For supersonic business jets, achieving good field performance will most directly conflict with achieving long supersonic range. Fortunately, achieving good field performance does not necessarily conflict directly with the low boom requirement because low wing loadings are beneficial from both perspectives. Nevertheless, stringent field performance requirements may still have an indirect impact on sonic boom because of their impact on the vehicle's weight.

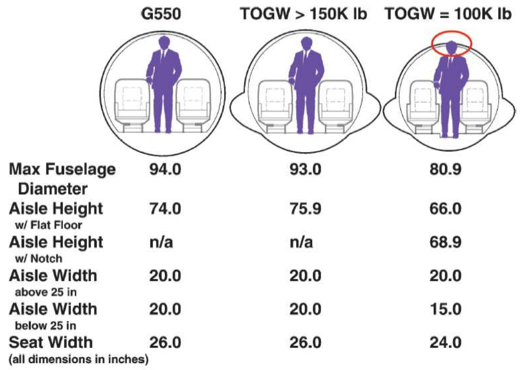
Community Noise: Current FAA regulations require to meet Stage IV guidelines, a 10 dB cumulative reduction over the Stage III guidelines. By the time a supersonic business jet enters service, even more stringent regulations may be in place. Meeting these noise constraints is even more difficult for supersonic vehicles than for conventional airplanes because supersonic engines have much higher exhaust velocities than corresponding subsonic engines. Although several approaches including complex Mixer-Ejector nozzles or thrust cutback can be used to meet the FAA regulations, they result in a significant weight and/or performance penalty.

5.1.3 Cabin comfort and volume

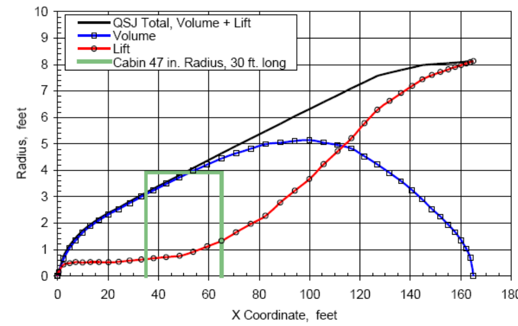
Configuration studies performed in [65] indicate that it will be difficult to integrate a large-diameter cabin into a 100,000 lb vehicle designed for supersonic overland flight. As shown in Figure 60, integration of large cabin diameters into a minimum-boom fuselage as determined by Seabass-George theory will be challenging. Like many other criteria, there is no clear consensus about what exact cabin size would be acceptable for a supersonic business jet. Although Gulfstream claims that a Gulfstream-II sized cabin is a requirement, others feel that a Citation X sized cabin would be sufficient considering the fact that trip durations will be much lower in a supersonic airplane.

5.1.4 Environmental impact

One of the only areas where the design of supersonic transports may be easier than that of their subsonic cousins is regulatory compliance with airport emissions. The



(a) Comparison of fuselage cross section for the subsonic G550 and two supersonic designs [65]



(b) Difficulty of integrating large cabins in lightweight low boom vehicles[72]

Figure 60: Issues with supersonic business jet cabin integration

relatively low pressure ratios used in supersonic engines result in low NO_x emissions in the vicinity of the airport. Although there are currently no regulations governing cruise emissions, this situation may change in the near future. [1] Aside from NO_x emissions, the impact of water vapor and CO_2 emissions can also be significant when released at altitude. Figure 61 shows that as the cruise altitude of the vehicle is increased above approximately 47,000 ft, the impact of the vehicle emissions increases rapidly. This effect may limit the maximum cruise Mach number of the vehicle because efficient high speed flight requires high cruise altitudes.

5.2 Problem Definition

The start of the proposed conceptual design process begins in much the same fashion as more traditional design efforts: identification of need and problem definition. Several firms have conducted market research into the economic viability of supersonic business jets, and have determined that a market exists for anywhere between 250 and 700 aircraft at an acquisition price of 65-100 million dollars each. [147]. In addition

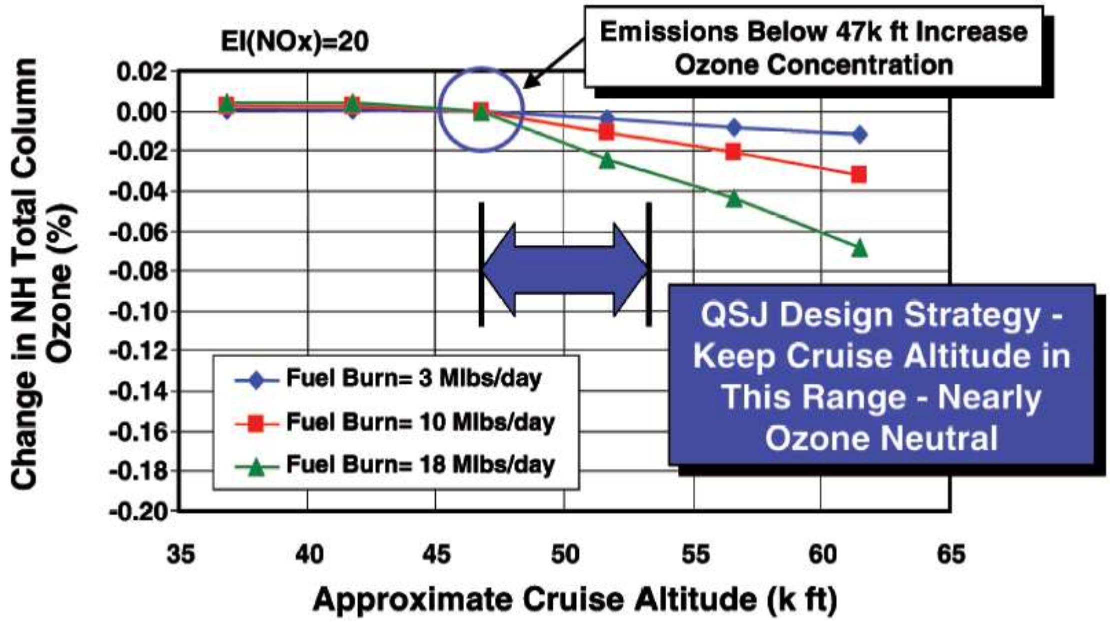


Figure 61: Impact of a fleet of supersonic business jets on the ozone as a function of cruise altitude [65])

to executive transportation, it is envisioned that such an aircraft would be useful for medical transportation, military, and airfreight applications. [20]

5.2.1 Requirements space definition

Despite the consensus on the marketability of a supersonic business jet, there is little agreement on what performance requirements are necessary for a successful program. Fortunately, the present method only requires ranges of desired capability rather than fixed requirements as described in Section 4.3.1.1. These ranges, which are presented in Table 10, were obtained by studying the requirements of published supersonic transport design studies by Gulfstream[161], Lockheed[94], Northrop Grumman[87], and others.

A few comments about these requirements and their values are in order. First,

Table 10: Proposed Small Supersonic Aircraft Requirements

Requirement	Objective
Range (nm)	4000-5000
Gross Weight(lbs * 1000)	100-120
Balanced Field Length (ft)	6000-7000
Cruise Mach Number	1.6-1.8
Payload	6-10 Passengers
Length (ft)	120-140
Static margin	0
Comfort	Cabin height $>$ 6ft, length $>$ 25ft
Takeoff/Landing Noise	FAA Stage IV
Sonic Boom	Acceptable for Overland Flight
Safety	Compliance with FAR Part 25
Acquisition Cost (Millions of USD)	65-100

airport noise constraints were not considered explicitly during the study because the propulsion systems used for this problem, which are described in Section 5.3.1.4, have both been fitted with Mixer-Ejector nozzles designed to comply with Stage IV regulations. Secondly, the nebulous nature of the sonic boom requirement is due to the fact that there is no law governing acceptable boom levels. Currently, the Federal Aviation Regulations prohibit overland supersonic flight. Although the FAA is currently investigating the possibility of modifying these regulations, no guidelines have yet been established. In order to proceed with the present study, "acceptable" has been defined as 84-88 dB PL at the start of cruise. Finally, there are several other important propulsion-related requirements that are not considered as a part of this proof of concept problem, such as a 2000 hour time between engine overhaul and low cruise NO_x emissions. A more detailed propulsion study will be required to quantify these metrics.

5.2.2 Concept space definition

Ideally, technology selection would be performed in parallel with concept and requirements exploration by including all technology options in the matrix of alternatives and then using the the concept exploration method to select the best technology portfolio. Unfortunately, technology benefit and degradation information was not available to the author, and instead it was decided to perform the proof of concept exercise with fixed technology assumptions appropriate for a vehicle intended to enter service in the next five to ten years. Individual values for these technology goals were determined by using the results of previous configuration studies, the 15-year supersonic sector goals from the GOTCHA charts of NASA's Vehicle Systems Program[27], and the expert input of several engineers at NASA. These values are listed in Table 11.

Table 11: Proof of concept technology assumptions

Metric	Delta
Specific Fuel Consumption	-5%
Wing weight	-18%
Fuselage weight	-5%
Empennage weight	-22%
Bare engine weight	-8%

The results of previous supersonic design studies were also used to populate a Matrix of Alternatives for the proof of concept problem. This matrix, presented in Table 12, contains 576 distinct configuration alternatives. Although this matrix is not comprehensive, the number of alternatives available is sufficient to act as a test problem for the concept exploration method.

In addition to the discrete alternatives listed in the Matrix of Alternatives, the

Table 12: Proof of concept Matrix of Alternatives

Planform Type	Double Delta	Ogee	Blended Arrow	Variable Geometry
Wing Location	Low	High		
Pitch Control	Horizontal Tail	T-Tail	Canard	Tailless
Engine Cycle	New mixed-flow turbofan	Variable cycle		
Airfoil	Biconvex	Reflexed	NACA symmetric	
Powerplant Installation	Under wing	Fuselage mounted	Tail mounted	
Inlet Type	Axisymmetric			
Inlet Compression	Mixed			
Inlet Geometry	Translating and collapsing			
Nozzle Type	Mixer-Ejector			
Materials	Mixed aluminum and composite			
High Lift System	Plain Flaps			

concept space is also defined by a large number of continuous design variables that control vehicle requirements and geometry. Some, like cruise Mach number of vehicle thrust-to-weight ratio are common to all configurations regardless of an individual's location in the design hierarchy. Other parameters such as wing planform kink locations are configuration-specific because these values are only relevant if a particular branch of the hierarchy is active. A structured encoding (see Section 2.3.5.3) was used to represent each configuration so that only pertinent continuous variables were used for evaluation, depending on the discrete settings.

5.3 Integration of the conceptual design simulation environment

In the course of his research on requirements and their influence on vehicle systems, Hollingsworth concluded that a new modeling environment capable of analyzing a large number of the possible system alternatives would be required to effectively search the matrix of alternatives of revolutionary design problems:

> The typical legacy conceptual design tool is a monolithic "black-box" that was developed to analyze a specific class or sub-class of vehicles. This inherently leads to several problems, not the least of which is that there may be no-one who truly knows all of the assumptions and simplifications that went into creating the tool. Compounding the problem that tools are designed for a specific type of vehicle, they are often calibrated with specific empirical data and relationships which do not hold for some sub-classes of the given vehicle type. This means that to study a truly diverse

Table 13: Proof of concept design variables

Category	Alternative	Parameter	Low	High
Fuselage		Length	120	165
		Cabin Location	0.25	0.4
		Cabin External Diameter	6.8	7.2
		Cabin Length	20	50
		Nose Droop Factor	0	0.05
		Nose Strength	0.6	1
		F-function balancing line slope β	-0.0002	0.0004
		Y_f	0.01	0.035
		Fraction of distribution	0.88	1
		Tail upsweep factor	0	1
Vertical Tail		Aspect Ratio	0.8	1.2
		Taper Ratio	0.4	0.7
		Volume Coefficient	0.06	0.1
		Sweep Angle [deg]	50	65
		Thickness to Chord Ratio	0.03	0.045
		Longitudinal Location of Trailing Edge**	0.92	0.97
Wing	Double Delta	Longitudinal Location of Apex**	0.6	0.72
		Vertical (Z) Location of Wing Plane	-0.85	-0.6
		Aspect Ratio	4	6
		Taper Ratio	0.1	0.3
		Wing Loading [psf]	50	75
		Sweep [deg]	40	65
		Dihedral (Root, Kink, Tip) [deg]	0	10
		Thickness to Chord Ratio Tip	0.022	0.025
		Thickness to Chord Ratio Root	0.03	0.035
		Wing Geometric Twist at Kink [deg]	0	2
		Wing Geometric Twist at Tip [deg]	0	5
		Strake Sweep [deg]	70	78
		Spanwise Bat-L.E. Wing Intersection **	0.25	0.5
		Spanwise Glove-L.E. Wing Intersection **	0.3	0.9
		Glove Aft Sweep [deg]	-15	15
	Ogee/Blended	Longitudinal Location of Apex**	0.5	0.65
		Vertical (Z) Location of Wing Plane	-0.85	-0.6
		Aspect Ratio	1.5	2
		Taper Ratio	0.02	0.08
		Wing Loading [psf]	45	70
		Glove Aft Sweep [deg]	-30	0
		Dihedral (Root, Mid, Tip) [deg]	0	10
		Thickness to Chord Ratio Tip	0.02	0.03
		Thickness to Chord Ratio Mid	0.025	0.04
		Thickness to Chord Ratio Root	0.025	0.04
		Wing Geometric Twist at Kink [deg]	0	2
		Wing Geometric Twist at Tip [deg]	0	5
		X1Mult(Bezier Control Pt.)	0	1
		Y1Mult(Bezier Control Pt.)	0	1
		X2Mult(Bezier Control Pt)	0	1
		Y2Mult(Bezier Control Pt.)	0	1
	Variable Geometry	Longitudinal Location of Apex**	0.5	0.7
		Vertical (Z) Location of Wing Plane	-0.85	-0.6
		Aspect Ratio	3.5	5
		Taper Ratio	0.45	0.65
		Wing Loading [psf]	60	90
		Thickness to Chord Ratio Tip	0.06	0.08
		Thickness to Chord Ratio Root	0.12	0.14
		Glove Taper Ratio	0.18	0.25
		Glove Span Multiplier	0.8	1.1
		Sweep (subsonic)	15	30
		Sweep (supersonic)	60	72
Horizontal Tail	None	None		
	Canard	Longitudinal Location of Apex**	0.1	0.2
		Aspect Ratio	1.8	2.5
		Taper Ratio	0.2	0.5
		Horizontal Volume Coefficient	0.1	0.16
		Sweep [deg]	45	60
		Thickness to Chord Ratio	0.03	0.05
	Conventional	Longitudinal Location of Trailing Edge**	0.92	0.97
		Aspect Ratio	1.6	2.2
		Taper Ratio	0.1	0.3
		Horizontal Volume Coefficient	0.2	0.4
		Sweep [deg]	45	60
		Thickness to Chord Ratio	0.03	0.05
	T-tail or Cruciform	Location on Vertical Tail Span**	0.6	1
		Aspect Ratio	1.6	2.2
		Taper Ratio	0.2	0.4
		Horizontal Volume Coefficient	0.2	0.4
		Sweep [deg]	45	60
		Thickness to Chord Ratio	0.03	0.05
Engine Pod	Under Wing	Spanwise Location of inboard Engine**	0.2	0.3
		Chord Wise Location of Engine Face**	0.4	0.7
	Fuselage	Location of engine face along body**	0.7	0.85
	Tail	none		
Control	Control	Cruise Lift Coefficient	0.08	0.12
		Supersonic Cruise Mach Number	1.6	1.8
		Take Off Gross Weight [klb]	95	135
		Thrust to Weight Factor	1.01	1.15
		** Denotes Fraction of Characteristic Length		

> set of systems and system types the engineer must collect and, in some manner, integrate several different tools, each with its own assumptions and limitations. [69]

At the start of this research, the intention was to rely upon highly simplified analysis methods to predict vehicle performance. However, it quickly became clear that the utility of the results would be much greater if more accurate models were used. This led to the development of a simulation environment that relies upon computer models that have gained acceptance by industry and government.

5.3.1 Integration of the disciplinary analyses

Since the advent of the digital computer, engineers have developed tools capable of analyzing and predicting the different aspects of vehicle performance, but the cryptic nature of the inputs and outputs has often made it difficult to integrate the tools for the purpose of multidisciplinary analysis. Instead, this integration was typically performed manually, with individual specialists running and interpreting the results of their own disciplinary codes. In recent years, several commercial frameworks including iSight [46] and Modelcenter [122] have been developed that facilitate communication between simulation models. These frameworks provide graphical user interfaces for creating and tracking information links between codes, and have been widely applied by both industry and academia.

Despite the availability of these commercial frameworks, the MATLAB programming environment was chosen for this study to handle inter-program communication, organize information flow, and to calculate performance metrics not provided by the

Table 14: Analysis tools used for the present study

Category	Name	Description
Geometry	Vehicle Sketch Pad[110]	Used to define the outer mold line
Aero.	AWAVE[63]	Wave drag based on linear theory
Aero.	BDAP[109]	Skin friction using wetted areas
Aero.	AERO2S[61]	Low speed induced drag w/empirical corrections
Aero.	WINGDES[61]	High speed induced drag and wave drag due to lift
Weights	FLOPS[104]	WTIN module within sizing code w/detailed wing weight calcs.
Sizing	FLOPS[104]	Calculates range and field performance using time integration
Boom	sgdarea[125]	Calculates minimum boom area distribution as $f(weight, mach, alt, length)$
Boom	PBOOM[26]	Calculates equivalent area distribution for sonic boom
Boom	pcboom4[123]	Calculates boom signature and loudness in dbA and dbPL

available disciplinary codes. MATLAB is a high-level programming language derived from FORTRAN that has been in development for over twenty years. [152] Although slower than compiled code such as C, it has gained popularity with the engineering community due to the large number of embedded functions that are available to solve statistics, signal processing, dynamic simulation, and other challenging problems. It can also be easier to write and prototype code in MATLAB than in other languages because it is an interpreted language.

Several criteria that influenced the decision to use MATLAB for the present study were: the familiarity most young Aerospace engineers have with MATLAB, the ease of creating graphical user interfaces within the environment, the ability to deploy codes written in MATLAB to other computers without a license, and the availability of distributed computing toolboxes that enable the program to make use of clusters of computers. In order to facilitate comparison with others' results, industry- and government-accepted disciplinary analysis were used where possible. (Table 14) When combined, these codes can be represented by a N^2 diagram as shown in Figure 62.

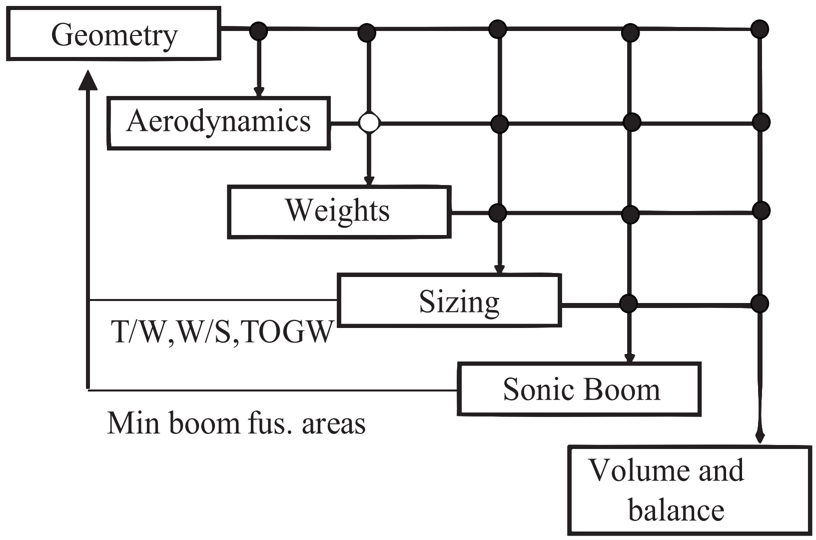
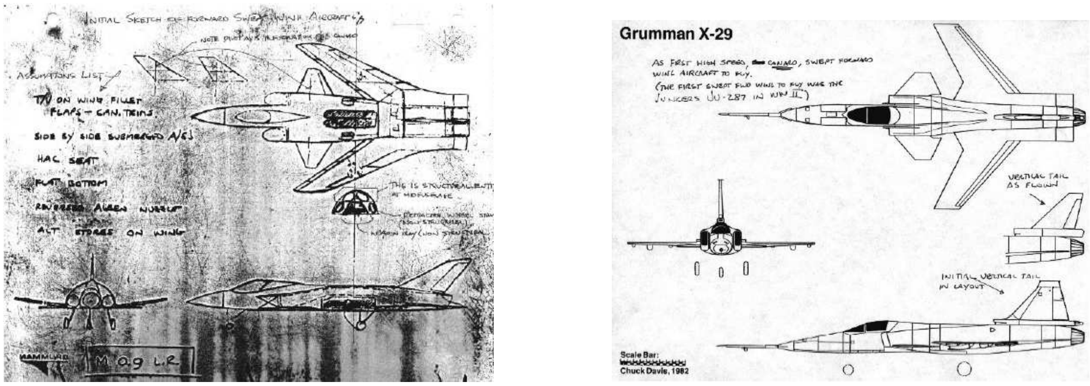


Figure 62: Schematic block diagram of the simulation environment

5.3.1.1 Geometry Representation

Geometry representation is an important consideration in conceptual design, as there is always a tradeoff between fidelity and the effort required to construct the description of the geometry. Historically, conceptual design often began with a rough "back of the envelope sketch" used to flesh out major vehicle features and the arrangement of major subsystems. As the design progressed, the sketch was iteratively improved with more and more detail. (Figure 63)

Computer Aided Design packages are now widely used by aircraft manufacturers in the development and production of new vehicles, yet often times the first steps of concept development are still done with pencil and paper. One of the primary reasons for this fact is that the process of describing an aircraft concept in traditional CAD packages may take significantly more time than a traditional sketch. [78] It has also been proposed that the rough nature of representations typical to conceptual



(a) (b)

Figure 63: X-29 (a) “Back of the envelope” sketch and (b) final configuration [86]

design can actually be beneficial because it promotes a free exploration of concepts and ideas. [50]

Commercial CAD packages were considered overkill for the present work because the conceptual designer is typically not concerned with structural details such as rivet placement or beam thicknesses. Instead, it is much more important to be able to describe concepts quickly and intuitively via parameters with which the designer is familiar.

The results of a search of available conceptual design geometry tools led to the selection of Vehicle Sketchpad (VSP) [110] as the geometry engine of choice for the present study. Vehicle Sketchpad is an updated version of Rapid Aircraft Modeler, a code developed at NASA to enable rapid prototyping, visualization, and analysis. One of the main advantages of the program is that engineers can describe vehicles in terms of design parameters with which they are familiar including sweep, area, and thickness-to-chord. Each representation is composed of a number of primitive shapes commonly used to represent aerospace vehicles such as “Wing” or “Fuselage”.

Within the program, a menu system can be used to easily add, reposition, or alter any of the primitives used to describe a vehicle concept. (Figure 64)

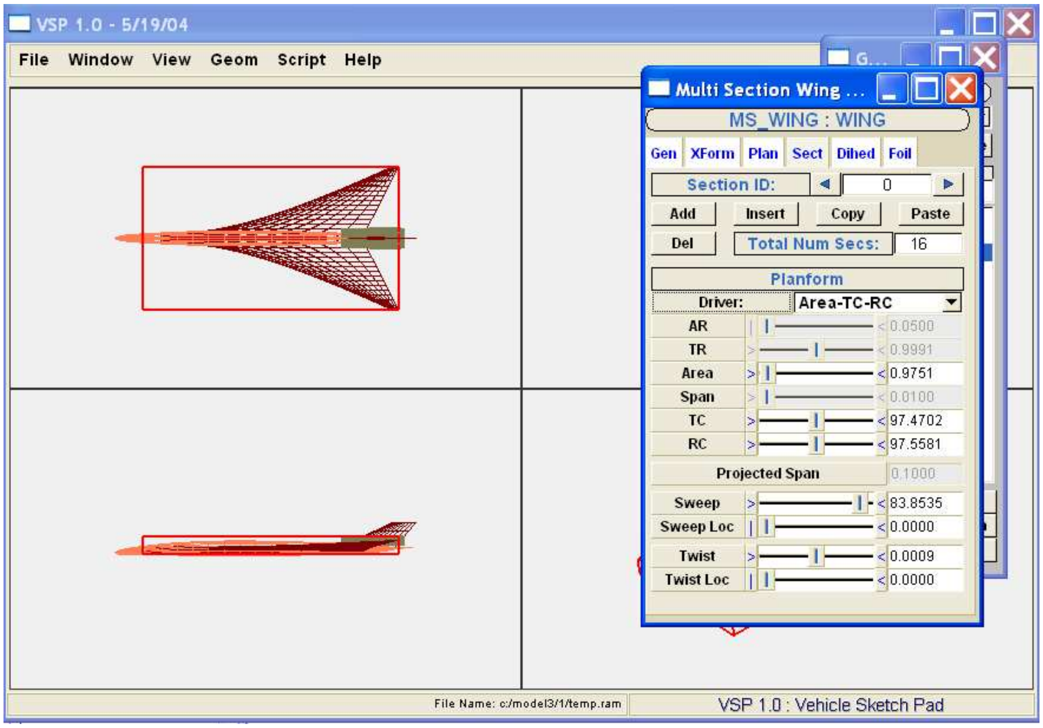


Figure 64: Screenshot of the Vehicle Sketchpad interface

This method of representation allows for the intuitive creation of vehicle models, and a simple configuration can often be created within minutes by a novice user. Although the geometry descriptions produced by VSP would not be adequate for detailed design or manufacturing, the ability to very quickly create and manipulate designs makes the tool appropriate for conceptual design use. VSP can write out geometry in a number of formats, including rhino3d, stereolithography, and Hermite. The ability to run the program in batch mode is also important for the present work because this facilitates incorporation into the automated modeling environment.

5.3.1.2 Aerodynamic Analysis

The requirement to evaluate thousands of designs during this concept selection research meant that processor time was a primary consideration, and eliminated the possibility of using sophisticated Euler or Navier-Stokes methods to calculate aerodynamic performance. Instead, a number of tools based upon linear theory were assembled to calculate properties such as supersonic wave drag, drag due to lift, and viscous drag.

Skin friction drag

The viscous drag of a streamlined aircraft configuration such as a supersonic transport is largely due to skin friction. The estimation of skin friction is a relatively simple procedure, and is handled by the Boeing Design and Analysis program, or BDAP. [109] This program computes the wetted areas and characteristic lengths of each of the aircraft components, and uses this information to compute the skin friction contribution of each assembly. The program assumes fully turbulent flow, which may be a slightly pessimistic assumption.

Interference and form factor corrections

The linear theory-based methods used for zero-lift drag calculations in the present study are incapable of estimating drag due to form factor and interference effects. Although these terms are only applicable at subsonic Mach numbers and will typically constitute less than $< 5\%$ of CD_0, their omission may still lead to over-optimistic subsonic performance predictions. Empirically-derived corrections from literature [127] are therefore used to account for these effects.

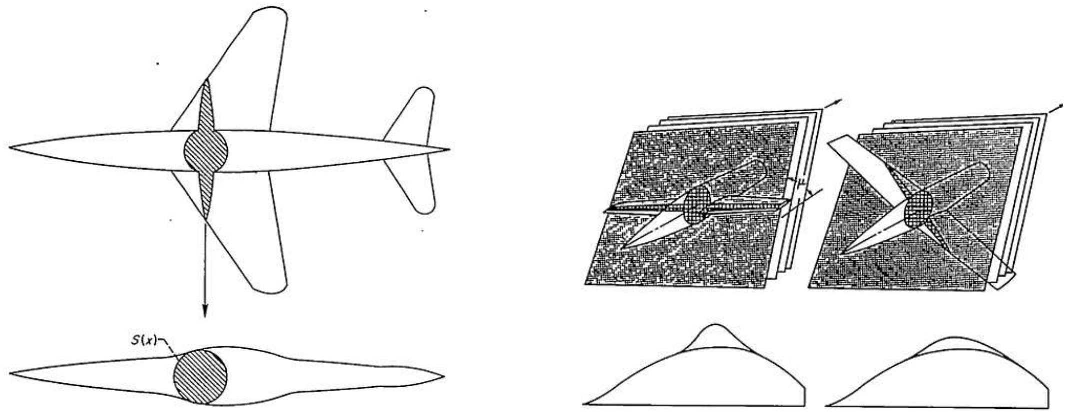
Wave drag due to volume

At high subsonic and supersonic Mach numbers, shock waves will form on an aircraft in flight. These shock waves will result in a substantial pressure drag on the vehicle, and in many cases may equal or exceed the sum of all other drag components combined. [157]

The wave drag of a slender, axisymmetric body in supersonic flow is given by Equation 14:

$$D_w = \frac{-\rho U^2}{4\pi} \int_0^l \int_0^l S''(x_1) S''(x_2) Ln|x_1 - x_2| dx_1 dx_2 \tag{14}$$

where ρ is the free stream density, U is the free stream velocity, S is the equivalent area as a function of axial position, and x_1 and x_2 are axial positions along the body. Linear aerodynamic theory [80] states that the wave drag of an arbitrary configuration may be estimated through the wave drag of its equivalent area distribution. At Mach 1.0, this representative body of revolution is found by taking a series of longitudinal cuts through the configuration and calculating the area at each station. (Figure 65(a)) For Mach numbers greater than unity, the area distribution must be calculated several times by a series of inclined planes at different roll angles. The total wave drag can then be calculated by averaging the wave drag of each area distribution calculated at different roll angle cuts. (Figure 65(b))



(a) (b)

Figure 65: Procedure for computing equivalent area at (a) Mach 1 and (b) supersonic Mach numbers

[80]

This procedure for estimating wave drag at supersonic speeds is incorporated into the conceptual design simulation environment through the Harris wave drag code, or AWAVE. [63] The use of AWAVE enables rapid wave drag calculation, but several limitations of the code and of linear theory should be noted.

One of these limitations is that due to the program's antiquated geometry input format, it is difficult to specify detailed aircraft geometry definition. This can lead to possible errors at component intersections. (Figure 66)

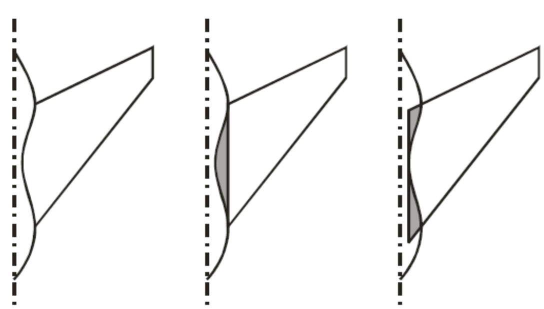


Figure 66: Poor component definition of AWAVE's geometry module [125]

Because of AWAVE's reliance upon linear theory, it cannot account for non-linear phenomena such as differential area ruling or shielding effects. [157] It also cannot calculate wave drag at subsonic speeds, which is primarily due to shock-induced separation. For this work, wave drag was calculated to be zero below a drag divergence Mach number obtained from Equation 15:

$$M_{DD} = \frac{M_{DD_{Airfoil}}}{\cos(\Lambda_{c/4})} \tag{15}$$

Above this M_{DD}, the wave drag is faired to the value computed by AWAVE.

Drag due to lift

Drag due to lift can be divided into three primary categories: vortex drag, wave drag due to lift, and profile drag due to lift. Vortex drag is also commonly referred to as induced drag, and for high aspect ratio wings can be accurately estimated using the very simple Oswald span efficiency method. [127] For more complex planforms, vortex drag can be reliably estimated by relatively simple vortex-lattice methods.

Wave drag due to lift arises from the pressure disturbances created by supersonic lifting bodies. Hayes [64] derived a linearized method for computing this contribution to drag through the concept of equivalent area due to lift. It was shown that this equivalent area can be calculated through Equation 16:

$$S(x) = \frac{\sqrt{M^2-1}}{2q} \int_0^x F_L' dx \tag{16}$$

where F_L' is the rate of growth of lifting force with distance. The wave drag due to lift of the configuration is equal is calculated by applying Equation 14 to the equivalent body of revolution defined by Equation 16 .

As previously mentioned, viscous separation effects at zero lift are accounted for through empirical form factor corrections. However, these corrections do not account for the additional viscous separation that results from adverse pressure gradients at non-zero angle of attack. The impact of this drag on the wing for wings of relatively low sweep and high aspect ratio can be accurately estimated through the use of simple sweep theory, 2-D airfoil section data, and span loading information. [15]

Accurate estimation of this contribution for the highly swept, low aspect ratio planforms typical of supersonic transports is more difficult because the theory of simple sweep is less easily applied. Fortunately, this contribution is negligible for properly designed aircraft at cruise, and relatively small for typical supersonic planforms at takeoff and landing, where vortex drag dominates. Therefore, viscous separation drag is not currently accounted for in the computational environment. Future enhancements to the code may include the addition of the capability to estimate this drag via 2-D section data and simple sweep theory.

For the present work, the AERO2S and WINGDES [61] codes were used to calculate the impact drag due to lift. These codes include a large number of empirical corrections for phenomena not accounted for by linear theory, including that of leading edge suction for airfoils with non-sharp leading edge radii.

AERO2S is used to calculate drag due to lift with and without flaps at subsonic speeds. The effects of horizontal tails or canards are also accounted for if present. WINGDES is used to compute drag due to lift at supersonic Mach numbers, but does not estimate the tail's contribution. This is not a significant issue because most SSTs are designed to be neutrally stable at cruise.

5.3.1.3 Weights Analysis

Weights analysis is still a difficult task for conceptual designers in the 21^{st} century. Though several codes such as ELAPS [146] and finite element methods have been developed for predicting structural weight in conceptual design, studies have not conclusively shown that the results of these codes are significantly more accurate than the much simpler methods based upon empirical weight equations and simple beam theory. [73] This is largely due to the fact that while these methods use advanced calculations to predict structural mass, they continue to rely heavily upon empirical non optimal mass factors (NOMFs). These corrections multiply the calculated structural weight to account for non structural components, and typically have an impact equal to or greater than the calculated structural contribution. [137]

Therefore, weights in the conceptual design environment are calculated by the WTIN routine within the Flight Optimization System (FLOPS). [104] This choice is beneficial because weight predictions are carried out nearly instantaneously, but it is recognized that a more detailed structural and weights analysis will need to be performed on the resulting concepts before proceeding to preliminary design.

5.3.1.4 Propulsion

The influence of the propulsion system on aircraft performance and viability is enormous because it has a large impact on fuel consumption, empty weight, and aerodynamics. The propulsion system interacts with the airframe, structure, and mission requirements in a complex fashion. It is therefore critical to design the propulsion system in parallel with the airframe in order to avoid negative interactions and to exploit favorable interactions where possible.

Unfortunately, for the present research it was not possible to perform a parallel design of the airframe and propulsion systems. Though the NASA Engine Performance Program (NEPP) and NOISIN noise estimation routine were integrated into the conceptual design environment, these analysis tools have been supplanted by the more capable but access-restricted Numerical Propulsion Simulation System (NPSS), Weight Analysis of Turbine Engines (WATE), and ANOPP. Based upon the advice of several propulsion specialists, all results for the present study were calculated using two fixed propulsion systems designed using these advanced tools.

> **Mixed-flow turbofan:** The first propulsion system used for the present research is an engine designed for cruise at Mach 2.0. It is shown in Figure 67 and consists of an axisymmetric translating centerbody inlet, a mixed flow turbofan engine, and a mixer-ejector nozzle with noise suppression.
>
> The nozzle has been sized to meet FAA Stage IV noise regulations. Spillage will occur when operating off-design, resulting in a thrust and SFC penalty for designs that cruise at less than the design Mach number of 2.0. The relatively

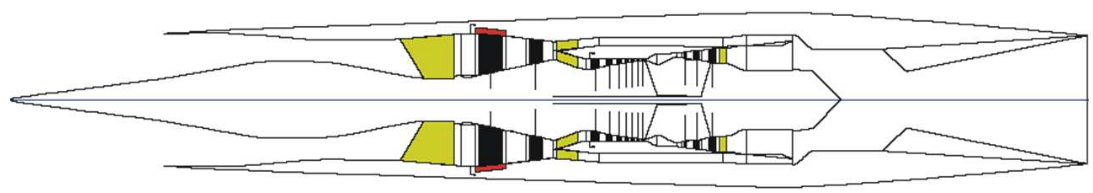


Figure 67: Mixed-flow turbofan flowpath

heavy and complex inlet used in the propulsion system is also an artifact of the high design cruise Mach number. This powerplant will therefore yield performance inferior to that of a specifically designed powerplant in the case where the vehicle cruises at a Mach number other than 2.0. The performance of the cycle is summarized in Table 15.

Table 15: Mixed-flow turbofan engine characteristics

Parameter	Value
Bare engine weight	4869 lb
Nozzle weight	3428 lb
Inlet weight	2012 lb
Mass flow at SLS	400 lbm/s
Thrust at SLS	22,205 lbf
SFC at SLS	0.678
Thrust at M1.8, 50,000 ft	6,501 lbf
SFC at M1.8, 50,000 ft	1.18

Variable Cycle:

The second propulsion system considered in the present study is a Variable Cycle Engine (VCE) designed for cruise at Mach 1.8. This system is depicted in Figure 68 and consists of an axisymmetric translating centerbody inlet, a variable cycle bare engine with a core-driven fan stage, and a mixer ejector nozzle.

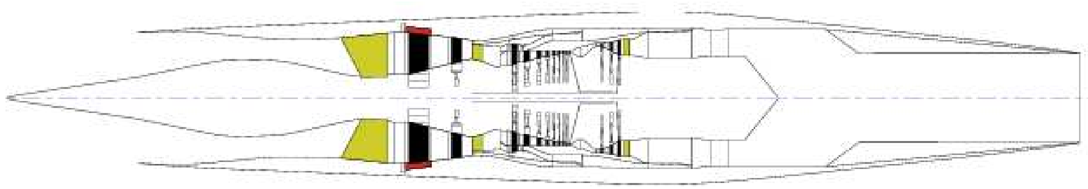


Figure 68: Variable-cycle engine flowpath

The mechanism for cycle variation on this particular engine is conceptually similar to that used on the F-120. One interesting aspect of the engine is that it produces more thrust at takeoff compared to the MFTF. Because of this, it may be possible to take off under part power, reducing noise and allowing for a smaller and lighter nozzle. (Figure 69) A summary of the performance of the VCE engine is given in Table 16.

Table 16: Variable-cycle engine characteristics

Parameter	Value
Bare engine weight	5622 lb
Nozzle weight	see Figure 69
Inlet weight	2012 lb
Mass flow at SLS	400 lbm/s
Thrust at SLS	23,074 lbf
SFC at SLS	0.72
Thrust at M1.8, 50,000 ft	5,439 lbf
SFC at M1.8, 50,000 ft	1.21

5.3.1.5 Stability and Control

Stability and control is an important consideration in aircraft design even today after the advent of fly by wire systems. If a vehicle is improperly balanced, there can often be severe performance penalties due to trim drag. This is especially true for supersonic vehicles due to the shift of aerodynamic center at supersonic Mach numbers.

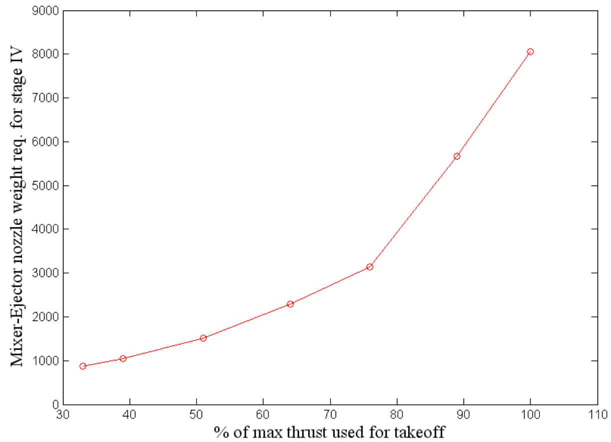


Figure 69: Impact of part-power takeoff on VCE nozzle weight

Although a detailed evaluation of stability derivatives, gust response, and handling qualities is outside the scope of the present study, it is critical to ensure that the vehicle will be controllable and have sufficient volume for fuel. One commonly used method for assessing these characteristics in conceptual design is through the use of center of gravity envelopes. These diagrams, an example of which is shown in Figure 70, depict the maximum forward and aft C.G. position achievable through fuel pumping as a function of aircraft weight.

The C.G. envelope can be used to evaluate longitudinal pitch and trim requirements through superposition of the vehicle's neutral point at critical operating conditions. If the neutral point lies within the C.G. envelope, then it will be possible to fly at that condition with no control surface deflection. Tip-back and rotation constraints can also be assessed by displaying the location of the main gear. Determination of the Center of Gravity envelope requires detailed knowledge of the fuel tank system and the location of every aircraft component. Although the position of

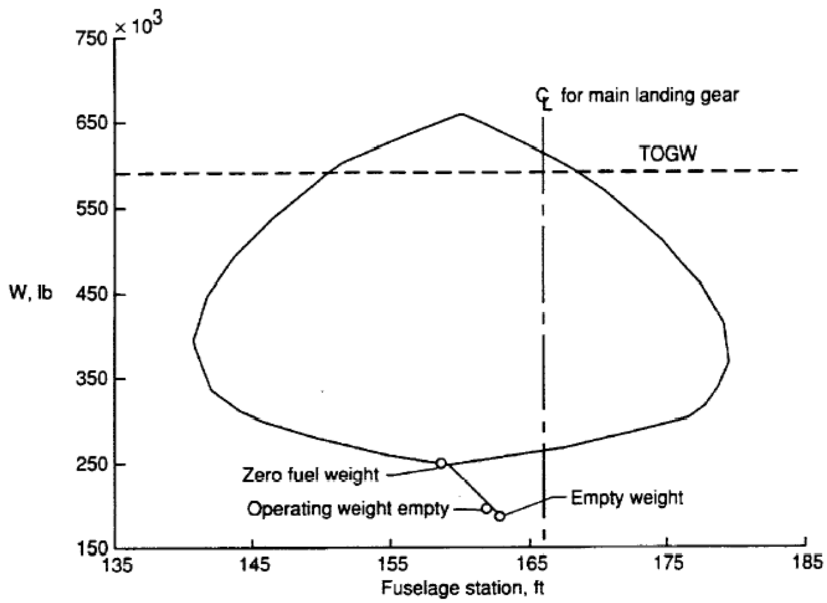


Figure 70: Example of a center-of-gravity envelope [139]

major structural members such as the wing or engines is known in conceptual design, the exact placement of other subsystems including air conditioning and electrical systems is typically not determined until slightly later in the design process. Historical guidelines (Table 17) were therefore used to estimate subsystem placement and to calculate the vehicle's C.G. envelope. [127]

Neutral point information at takeoff, landing, top of climb, and end of cruise is calculated by AERO2s and WINGDES. At low speeds, AERO2S estimates the contribution of flap and horizontal tail deflection. For supersonic cruise, it is assumed that tail incidence will be set to produce zero lift. Transonic stability is an important factor not considered in the present study because the tools being used are not capable of predicting the unsteady and non-linear phenomena associated with this flight regime.

The neutral point and center of lift information is used in conjunction with the

Table 17: Guidelines used to calculate component placement

Component	Location
Wing	40% wing MAC
Horizontal Tail	50% H.T. MAC
Vertical Tail	40% V.T. MAC
Fuselage	50% fuselage length
Nose Gear	20% fuselage length
Main Gear	65% wing MAC
Engines	66% engine length (from fan face)
APU	40% V.T. MAC
Flight Crew	cabin location -5 ft
Passengers	center of cabin
Furnishings	center of cabin
Wing Fuel	as calculated by AWAVE
Fuselage Fuel	as calculated by AWAVE

center of gravity envelope to construct a balance and volume figure of merit used as an objective during design and optimization. This figure of merit is formulated as the difference in area calculated between a hull containing the vehicle's centers of lift and its C.G. envelope (Figure 71). Minimization of this figure of merit ensures that the vehicle has sufficient volume for fuel and can be trimmed at all steady flight conditions. Transient effects such as transonic acceleration are not accounted for, and these effects should therefore be examined in detail before proceeding with the design.

5.3.1.6 Mission Analysis

Simple mission analysis calculations can be performed through the use of the Breguet range equation and other textbook methods. However, a more detailed and accurate assessment of critical requirements including range and takeoff performance can be obtained through time integration for relatively little computational effort.

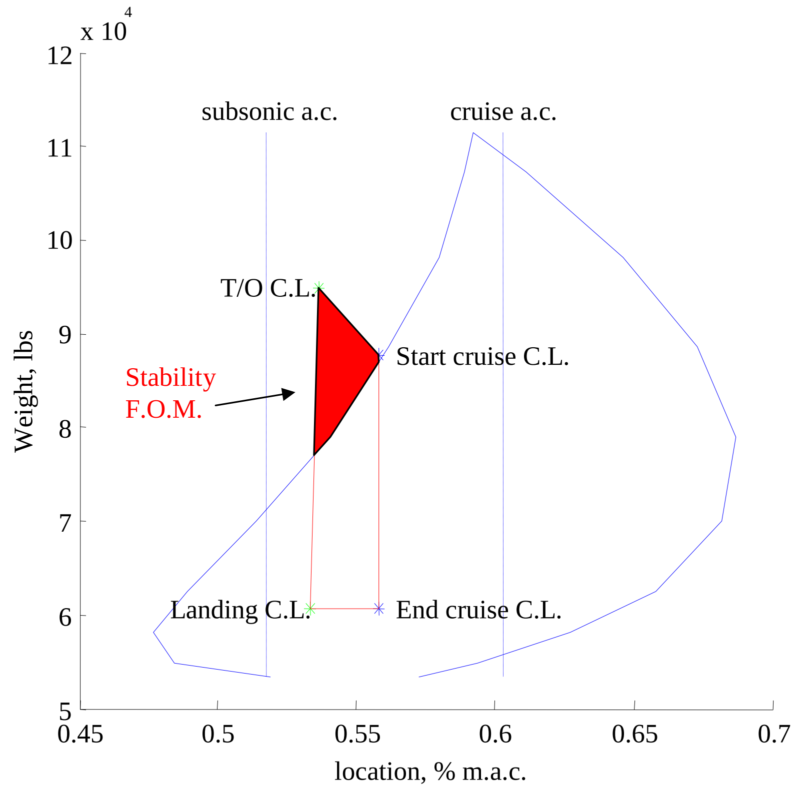


Figure 71: Figure of merit used to assess longitudinal control and volume availability

Several multidisciplinary computer codes capable of performing these calculations are available, including FLOPS and ACSYNT. [104] These codes are often referred to as monolithic, because they are also capable of analyzing many aspects of vehicle capability such as aerodynamics, propulsion system performance, and cost. For the present study, FLOPS was used only to calculate vehicle weight and to perform mission analysis.

5.3.1.7 Sonic Boom Analysis

The theory used to derive linearized sonic boom prediction is closely related to that of the linearized supersonic drag prediction methods developed by Hayes. In 1952, Whitham proposed the method of linearized sonic boom prediction that relies upon representations of lifting configurations as equivalent bodies of revolution, or "equivalent areas." [158] This equivalent area is the summation of the configuration volume used to calculate zero lift wave drag and equivalent area due to lift used to calculate wave drag due to lift. (Figure 72)

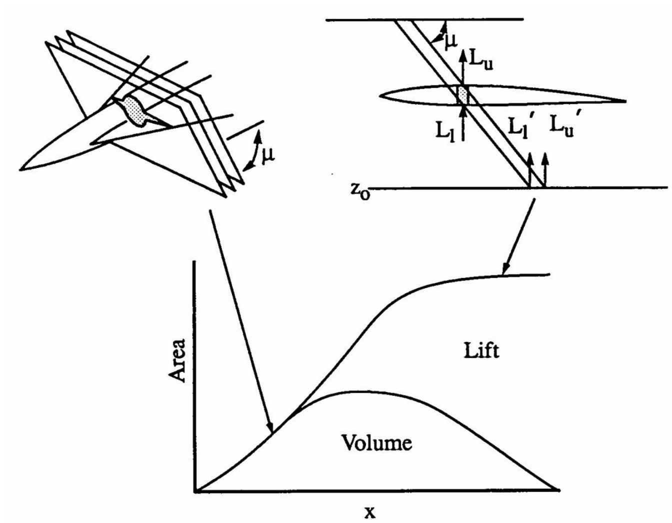


Figure 72: Formulation of the equivalent area concept [26]

It was shown in [158] that if the equivalent area was known, the pressure disturbance of the configuration could be calculated through Whitham's "F function," Equation 17:

$$F(y) = \frac{1}{2\pi} \int_{0}^{y} \frac{S''(t)dt}{\sqrt{y-t}} \tag{17}$$

Several early methods based upon characteristic ray tubes and the waveform parameter method were developed that predicted signature strength when propagated to the ground. [156][153] Recent research has indicated that nonlinear atmospheric effects can have a significant impact on the propagated signature, and Plotkin [123] has developed an improved propagation code known as PCBOOM that is now widely used and is capable of estimating the impact of some of these nonlinearities.

For the present study, a tool known as PBOOM [26] was used to calculate the F-function for a given vehicle configuration and PCBOOM4 [123] was used to calculate the propagated ground signature. Because the signatures produced by the propagation procedure are jagged and do not resemble real world data, hyperbolic tangent smoothing was applied to the signature before the calculation of dbA and dbPL loudness metrics.

5.3.2 Environment verification

In order to assure that the tools and methods being used for the work are consistent with those used by other researchers for supersonic aircraft analysis, a supersonic multi-mach aircraft geometry provided by engineers at NASA Langley was independently analyzed using the modeling environment. The reference geometry was provided in the Craidon format used by AWAVE [29]. The features of this vehicle were extracted from the geometry file by MATLAB scripts, and were used as inputs to the present environment. Rather than serve to validate the tools versus real world data, this verification process was used to uncover logical errors in the code and to ensure the assumptions used by the model are reasonable.

The reference vehicle shown in Figure 73 is based upon a concept developed during the High Speed Research program. It has been resized to carry 175 passengers over a 5500 nm 50% Mach 0.95, 50% Mach 2.0 mission [138]. The powerplant was also supplied by NASA and includes a mixer-ejector nozzle sized to meet FAA stage III noise

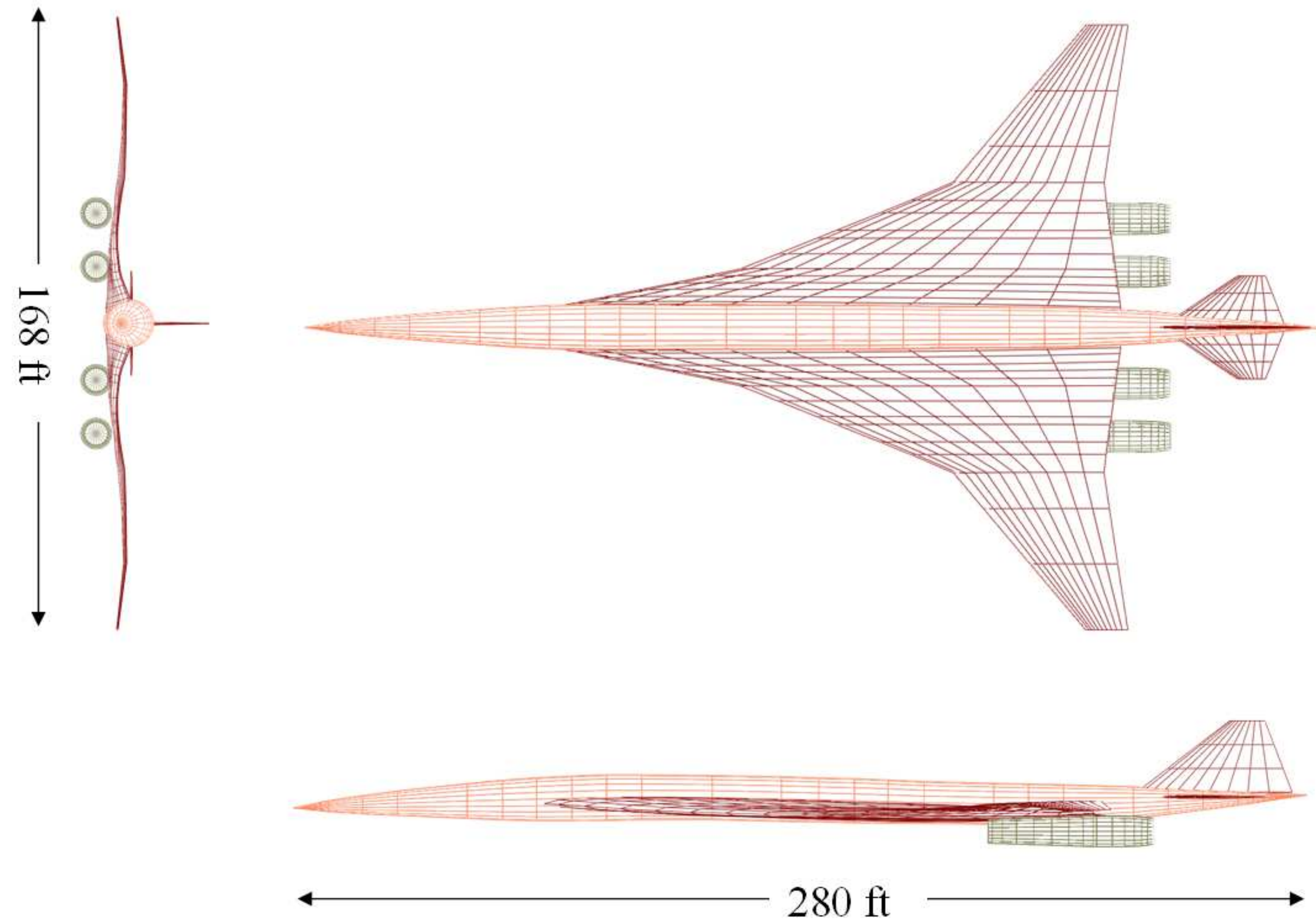


Figure 73: Three-view of the supersonic reference vehicle provided by NASA

levels. In order to simulate present day composite technology levels, the technology K-factors listed in Table 18 were applied to the weights computed by FLOPS. These same factors were used to generate the NASA-supplied results. In order to obtain results for comparison, the present modeling environment was run in analysis mode with the same gross weight as that of the supplied reference vehicle.

Table 18: Technology K-Factors used for Environment Verification

Technology Factor	Value
Wing Weight	0.75
Fuselage Weight	1.05
Empennage Weight	0.85

5.3.2.1 Weights comparison

One of the most influential metrics on vehicle performance is empty weight fraction, and it is therefore very important to ensure that weight estimation results are realistic. In Table 19, the results from the previous analysis are compared with those calculated during the current work. On the whole, the predicted weights are similar, with an empty weight difference of about 3% between the two models. Individual subsystems, however, exhibit more variability and merit additional analysis.

Within the structural weights category, wing weight and landing gear weight vary significantly from each other. In fact, the difference in predicted wing weight between the two models accounts for the majority of the total empty weight discrepancy. After consulting with engineers at NASA, it was determined that the weight of the reference model had been calculated using 3/4 chord sweep values. The present model instead uses trailing edge sweep values as recommended in [52]. This source of discrepancy was verified by re-running the present model using 3/4 chord sweeps, but it was decided to retain the use of trailing edge sweep for future analysis. The difference in landing gear weight was investigated, and found to arise from the fact that the current model is designed using a slightly higher maximum landing weight.

The difference in propulsion system weight stems solely from the use of different engine scaling exponents, 1.0 for the current study versus 1.15 for the previous analysis, since the engines and thrust levels are identical for the two vehicles. There are several other non structural items such as avionics that vary significantly between the models in a percentage bases, but their absolute difference is small and was not investigated further. Overall, it appears that with the exception of the decision to use trailing edge sweep values for the load path, the current results largely agree with those that were previously generated by NASA.

5.3.2.2 Aerodynamics Comparison

Because the linear aerodynamic tools used for both the present study and for the NASA results are identical, the aerodynamic performance predicted by the current model should be very similar to that of the previously generated results. This was in fact generally the case (Figures 74:76). In some cases, there were some minor differences such as in subsonic zero lift drag. This is likely due to the fact that the drag polars used in the reference configuration were scaled within FLOPS, while those used in the present analysis were not. Other minor differences may also be due to the use of different empirical corrections such as that used to account for subsonic form factor drag.

Table 19: Weights Analysis Comparison

MASS AND BALANCE SUMMARY	Calculated	Reference	% Difference
WING	74621	84278	-11.46%
HORIZONTAL TAIL	2952	3084	-4.28%
VERTICAL TAIL	2375	2385	-0.42%
FUSELAGE	49996	49996	0.00%
LANDING GEAR	29998	27908	7.49%
NACELLE (AIR INDUCTION)	0	0	
STRUCTURE TOTAL	159942	167652	-4.60%
ENGINES	87339	89438	-2.35%
THRUST REVERSERS	0	0	
MISCELLANEOUS SYSTEMS	1463	1463	0.00%
FUEL SYSTEM-TANKS AND PLUMBING	5443	5026	8.30%
PROPULSION TOTAL	94245	95926	-1.75%
SURFACE CONTROLS	9132	9130	0.02%
AUXILIARY POWER	1192	1192	0.00%
INSTRUMENTS	1366	1546	-11.64%
HYDRAULICS	5143	4301	19.58%
ELECTRICAL	4399	4542	-3.15%
AVIONICS	1983	2718	-27.04%
FURNISHINGS AND EQUIPMENT	18987	19825	-4.23%
AIR CONDITIONING	5449	5504	-1.00%
ANTI-ICING	483	363	33.06%
SYSTEMS AND EQUIPMENT TOTAL	48134	49121	-2.01%
WEIGHT EMPTY	301779	312700	-3.49%
CREW AND BAGGAGE-FLIGHT, 2	450	675	-33.33%
-CABIN, 4	665	975	-31.79%
UNUSABLE FUEL	1574	1896	-16.98%
ENGINE OIL	367	367	0.00%
PASSENGER SERVICE	2681	3318	-19.20%
CARGO CONTAINERS	1575	1575	0.00%
OPERATING WEIGHT	309091	321506	-3.86%
PASSENGERS, 175	28875	28875	0.00%
PASSENGER BAGGAGE	7700	7875	-2.22%
ZERO FUEL WEIGHT	345666	358256	-3.51%
MISSION FUEL	468232	455642	2.76%
RAMP (GROSS) WEIGHT	813898	813898	

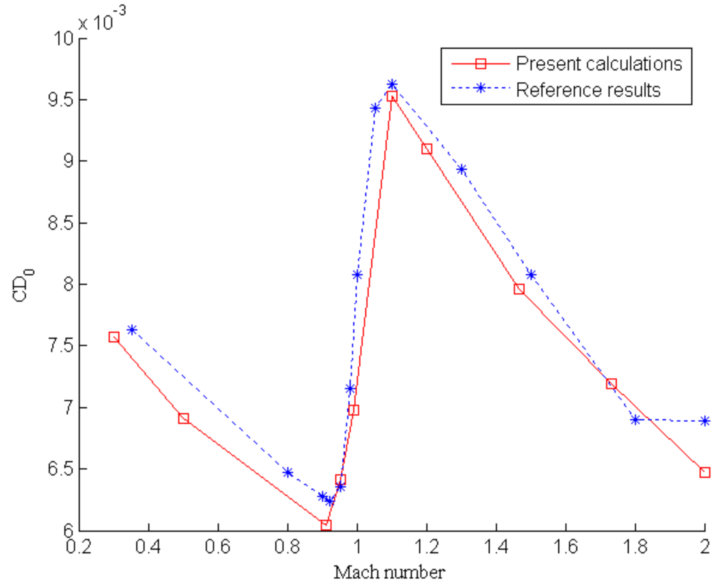


Figure 74: Zero lift drag comparison at 50,000 ft

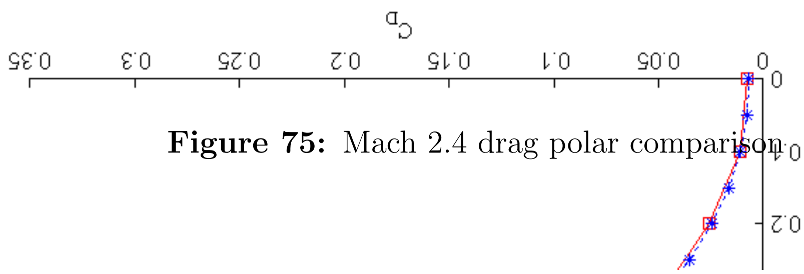


Figure 75: Mach 2.4 drag polar comparison

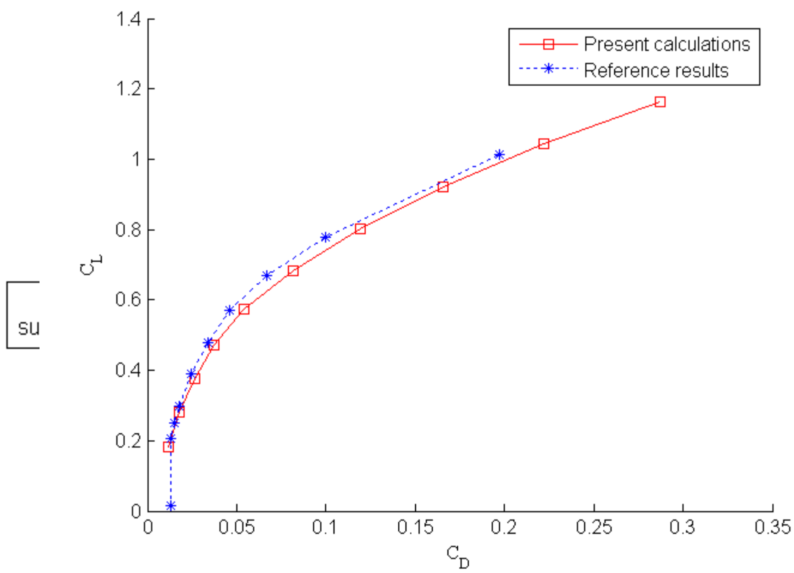


Figure 76: Takeoff drag polar comparison

5.3.2.3 Sonic boom prediction verification

The sonic boom prediction methods used for the present research rely upon linear theory, which has existed for decades but was only validated very recently. Under the DARPA QSP program, Northop Grumman modified an F-5E aircraft with a nose glove to achieve a shaped sonic boom signature. [121] Several test flights in late 2003 demonstrated that shaped signatures can in fact be achieved.

The sonic boom analysis tools used for the present study (PBOOM and PCBOOM) were actually used during the design of the shaped boom demonstrator. [121] In [49], this boom demonstrator model and that of an available F-5A were analyzed using PBOOM. Although the geometry of the shaped boom demonstrator's nose glove is proprietary, a representation of the geometry was created from publicly available photographs and Rapid Aircraft Modeler.

The sonic boom signatures of the modified and unmodified F-5s were predicting using the published test flight conditions and PBOOM. It was shown that PBOOM accurately predicted the unmodified F-5's N-wave signature, and also did a reasonable job of predicting the shaped boom of the modified aircraft. The author of [49] suspected that the differences between actual and predicted rear shock strengths may be partially due to the fact that an F-5A model was used rather than the F-5E that actually flew.

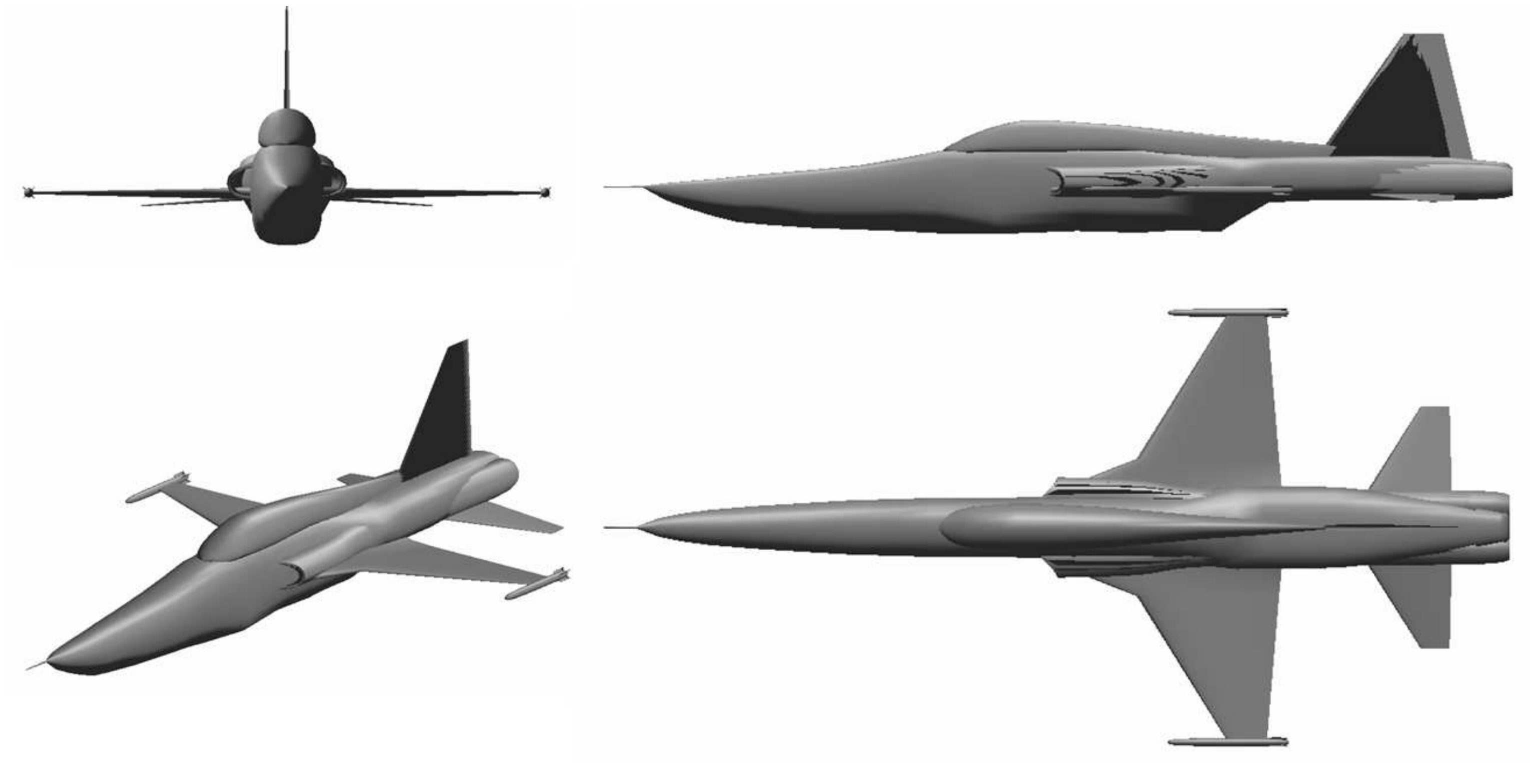
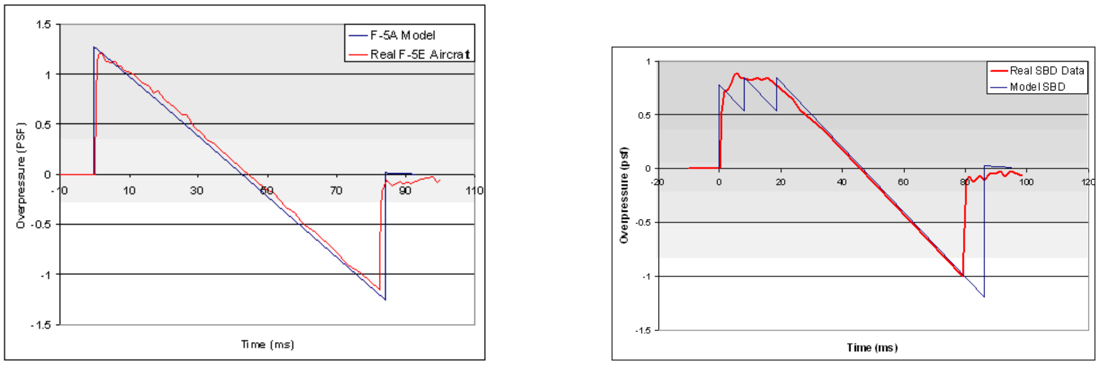


Figure 77: Three-view of the shaped boom demonstrator model used for Pboom validation [49]



(a) (b)

Figure 78: Measured and predicted ground signatures for the (a) unmodified F-5 and (b) shaped boom demonstrator
[49]

5.4 Concept space exploration using multi-value optimization

Only ten of the requirements listed in Table 10 were actually used as objectives in the analytical portion of the proof of concept problem, which is defined in Table 20. Other requirements, such as payload were addressed via constraints, while safety, acquisition cost, and other issues were left unaddressed for the time being.

Table 20: Multi-value optimization problem definition

Maximize:	range
	cruise Mach number
	fuselage diameter
Minimize:	sonic boom loudness
	shock pressure rise
	gross weight
	balanced field length
	approach velocity
	balance metric
	length
By varying:	configuration (Table 12) and decision variables (Table 13)
Subject to:	side constraints (Table 13)

This optimization problem was represented using the method of hierarchical encoding described in Section 4.2, and solved by using the parallel Multi-value Genetic Algorithm and the settings listed in Table 22. The Multi-value Genetic Algorithm requires the decision-maker to specify vectors of goal and threshold values corresponding to ideal and acceptable levels of performance in each numerical objective. These vectors, given in Table 24, were populated using the program objectives established earlier in Table 10. The normalization constants required by the algorithm were determined by calculating the variance of each objective for a random population of individuals and then rounding that value to a convenient number.

Table 21: Goal and threshold levels used by the MVGA value functions

Objective	Goal	Threshold	normalization constant
Range (nm)	5000	4000	2
Shock pressure rise (psf)	0.3	0.4	0.002
Takeoff field length (ft)	6000	7000	5
Approach velocity (kts)	130	140	0.2
Boom loudness (dB PL)	84	88	0.01
Cruise Mach number	1.8	1.6	0.0025
Gross weight (lb)	100,000	120,000	100
Balance metric	1	100	5
Length (ft)	130	140	0.05
Fuselage diameter (ft)	7.0	6.8	0.01

The Multi-value Genetic Algorithm was executed using 75 nodes of a high performance 256 CPU computer cluster operated by Georgia Tech. Each node consists of two 3.2 GHz Pentium IV processors and is equipped with four gigabytes of memory. Total execution time was just under twelve hours.

Table 22: Parameter settings for supersonic business jet concept exploration problem

Parameter	Main population	Secondary populations
Population size	600	200
Crossover rate	70 %	70 %
Crossover operator	Structured Crossover	(Algorithm 1)
Mutation rate (continuous)	2%	2%
Mutation rate (discrete)	2%	2%
Mutation type	uniform	uniform
Mating similarity parameter α	1	1
Mating similarity parameter β	3	3
RTR window size ω	$600 - 10m$	200
Termination criteria	200,000 total function calls	50 generations

5.4.0.4 Investigation of Multi-value Genetic Algorithm results

One of the drawbacks of using the Pareto multiobjective optimization approach for problems with more than 2 or 3 objectives is the difficulty associated with visualizing the results. Modern displays are typically limited to two dimensions, leading to the necessity to view projections of the calculated Pareto hypersurface. This is most commonly done through the use of a scatterplot diagram that shows every pairwise projection in a matrix form. The results from the proof of concept Multi-value optimization procedure are presented in scatterplot form as Figure 79.

Unfortunately, these results remain difficult to interpret because the data appears as a "cloud" rather than a line or a surface. It can also be difficult to determine the relationship between the individual plots in the scatterplot matrix. The value-path method suffers from many of the same difficulties. One method of enhancing the usefulness of the scatterplot matrix for problems with many objectives is to color the different data projections according to the degree of pairwise conflict. An example of

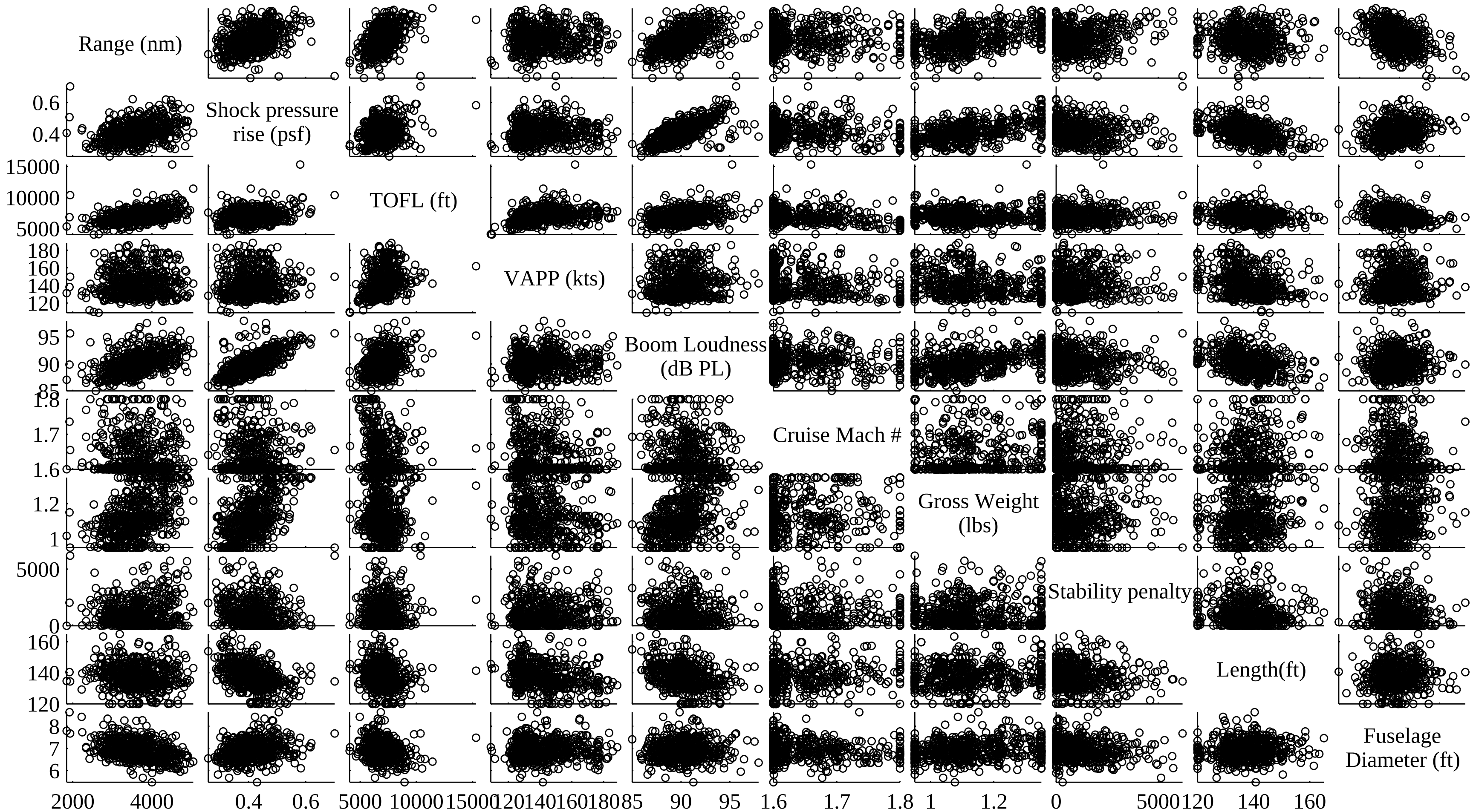


Figure 79: Scatterplot representation of the 10-D Proof of Concept Pareto Hypersurface

this type of scatterplot is presented as Figure 80, where beneficial relationships have been colored green and conflicting relationships have been colored red. The intensity of the relationships is represented by the intensity of the color.

Although scatterplots can help to clarify the relationship between conflicting requirements, they do not provide insight into the relationship between requirements and preferred system concepts. In order to investigate these effects, a visualization method called the s-Pareto frontier which was described in Section 2.4 has been used. A large number of s-Pareto frontier projections have been assembled in Appendix C, and some of the more interesting relationships are examined here in more depth.

The first of these relationships is presented in Figure 81. This figure depicts the tradeoff between takeoff field length and range, and is colored according to the vehicle's wing planform type. In this projection, the variable geometry concept clearly exhibits superior performance compared to the three fixed geometry concepts which are also represented in the figure. Although this is hardly a surprising result, it is still interesting to examine the amount of range that must be sacrificed by the fixed wing concepts to achieve short field performance.

The impact of the pitch control mechanism on the takeoff field length and range was also investigated using the results presented in Figure 82. One of the first things observed is that although conventional tails were included as an alternative in the design problem, nearly none of these designs survived to become members in the final Pareto optimal set. After performing an investigation into this effect, it was found that these designs were severely penalized by the sonic boom requirement. Because it is desirable to place the wing relatively far aft in order to maximize usable volume,

Figure 80: Scatterplot of the Pareto Hypersurface colored using the correlation matrix

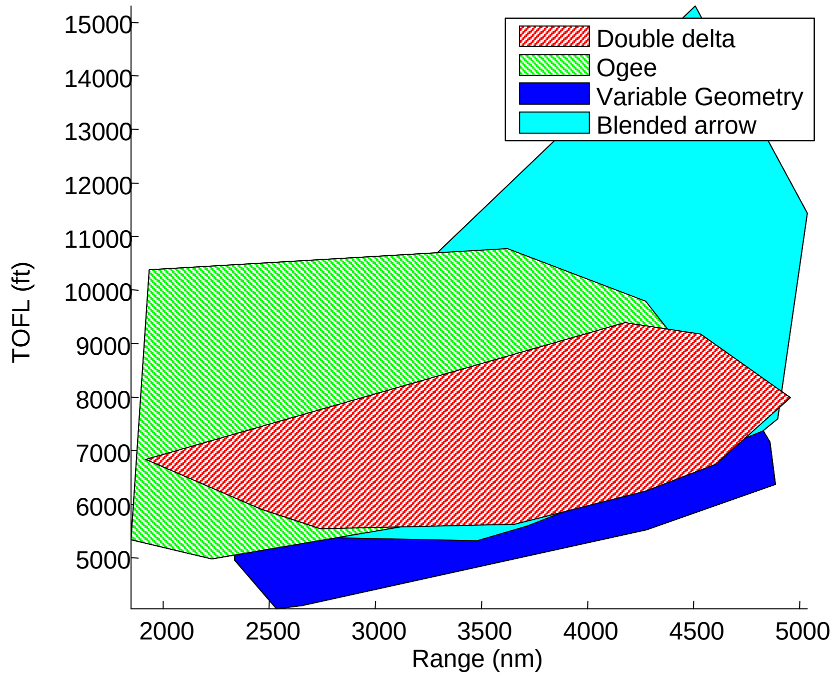


Figure 81: Impact of planform on range and takeoff field length

designs with conventional tails tended to have very short tail moment arms, resulting in very large tails and excessive wave drag. Examination of Figure 82 also reveals that tailless concepts are preferable with respect to range capability, while airplanes with tails exhibit improved takeoff performance. This result was not surprising, because the vehicles with tails are able to make greater use of flaps than tailless designs.

The effect of the control surface configuration on range and boom loudness was also investigated by using Figure 83. The tailless vehicles had much greater range than the vehicles with control surfaces due to reduced empty weight and skin friction, but the reason behind the observed boom loudness superiority of the T-Tail concepts was not as obvious, especially since the control surfaces were assumed to have zero load during cruise. In order to investigate this phenomenon, the overpressure signatures

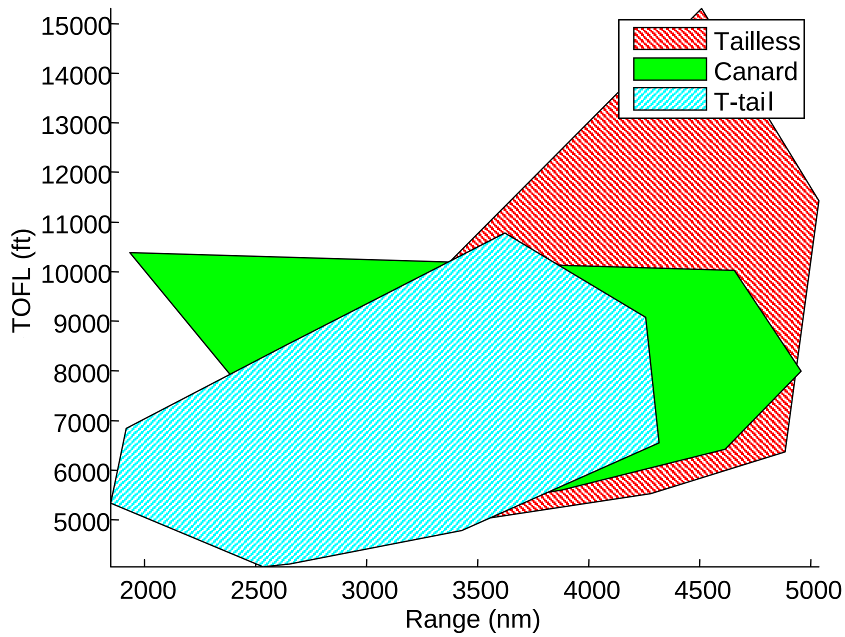


Figure 82: Impact of control surface arrangement on range and takeoff field length

and equivalent area distributions of a T-tail and a tailless configuration were plotted in Figure 86. This study revealed that even neglecting the contribution of tail lift, the T-tail provides off axis body addition that significantly extends the equivalent length of the vehicle and softens the rear shock as shown in Figure 85, thereby reducing the loudness. Unfortunately, Figure 83 also shows that there is a very significant range penalty associated with the use of a T-tail.

Another interesting relationship was revealed through examination of the impact of airfoil type on range and boom loudness as displayed in Figure 87. The author's intuition had indicated that the latter two types of airfoils represented in Figure 87 might be easier to incorporate in a low boom vehicle, but the results of the Multi-value Genetic Algorithm indicate that this is not necessarily the case. This significant

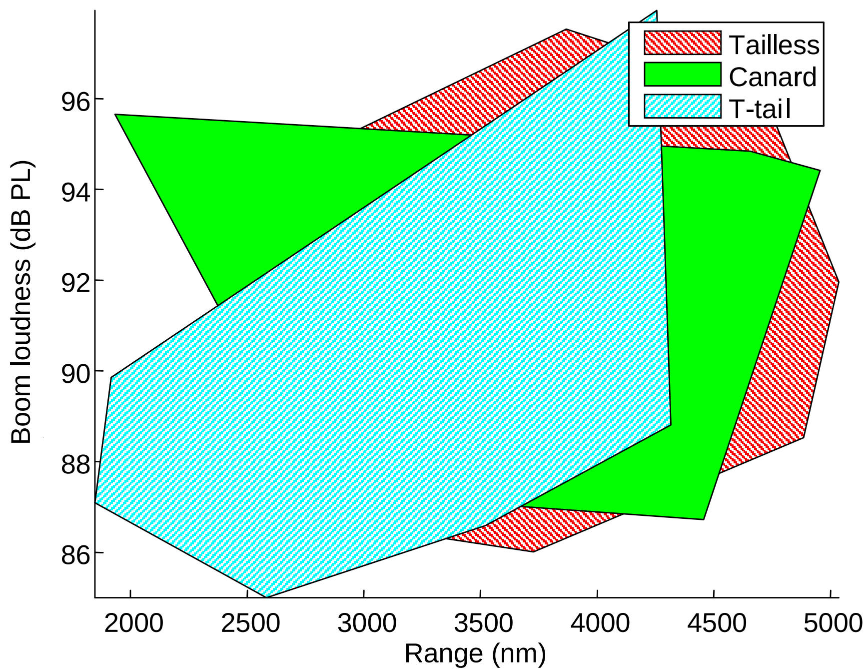


Figure 83: Impact of control surface arrangement on range and boom loudness

impact of airfoil selection on these metrics was also surprising when considering the linearized Mach-box method used to calculate the equivalent area distribution due to lift. In fact, while the calculated lift distribution was dependent only upon the planform, the contribution due to volume was strongly dependent on airfoil type. Examination of the wing's volume distribution for the three different airfoil types revealed that as shown in Figure 88, both the biconvex and aft-loaded and reflexed sections had significantly higher maximum cross-sectional areas in the $\theta = -90$ slice used to calculate sonic boom below the aircraft. When integrated into a low-boom area distribution, these distributions led to higher drag.

Figure 89 displays the tradeoff between range and gross weight as a function of the engine cycle. Somewhat surprisingly, the Variable Cycle Engine performs much worse than its Mixed-Flow Turbofan counterpart. After consultation with the

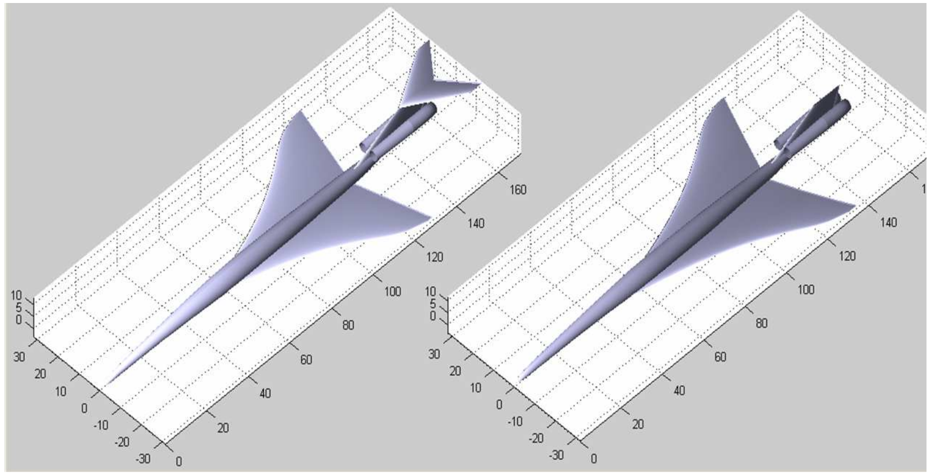
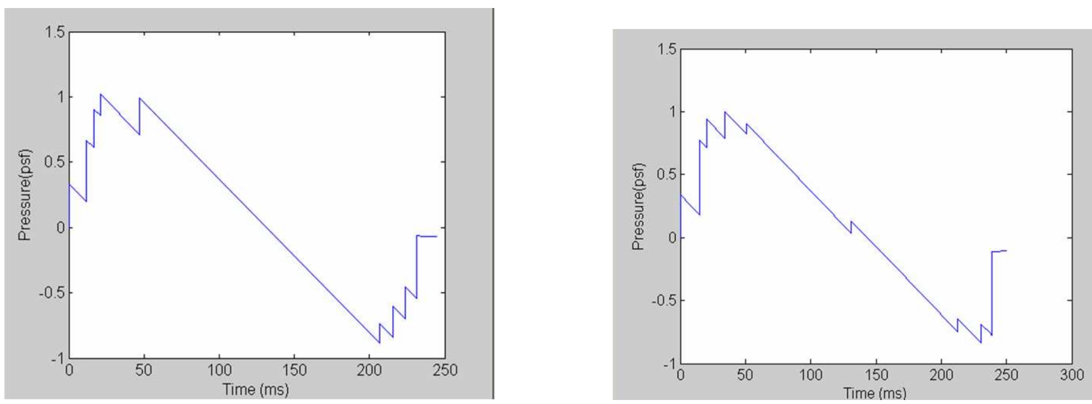


Figure 84: Example Low-Boom T-Tail and Tailless Configurations

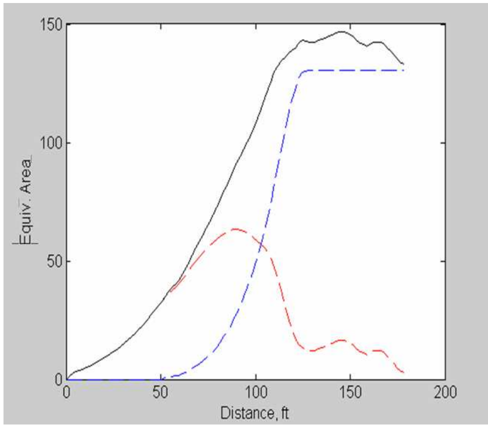


(a) (b)

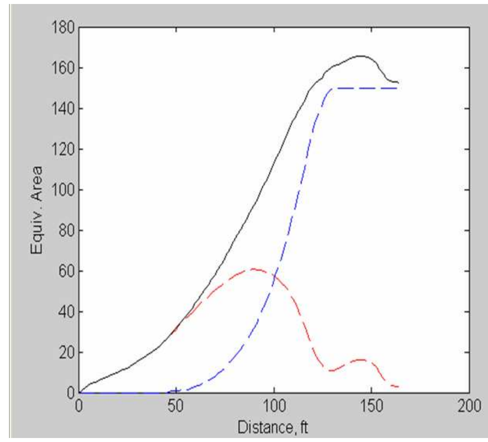
Figure 85: Sonic Boom Signature for (a) T-Tail and (b) Tailless Configurations

propulsion systems analyst that created the cycle, it was determined that unlike the Mixed Flow Turbofan, the Variable Cycle Engine's cycle parameters had not yet been optimized. Additionally, the engine was intended for use on a vehicle with a 50/50 subsonic/supersonic mission split. Because of these factors, the propulsion system comparison is probably not a fair one.

The static nature of these s-Pareto frontier projection prevents the designer from exploring more complex multi-factor interactions. This issue has been addressed by



(a)



(b)

Figure 86: Equivalent Area Distribution for (a) T-Tail and (b) Tailless Configurations

developing an interactive results exploration tool, pictured in Figure 90. By using this tool, the decision maker can select projections of the Pareto hypersurface and then dynamically view the effect of imposing constraints in the other inactive dimensions. The data points can be color coded according to configuration type, providing similar information as the static s-Pareto frontiers. Additional information about a particular data point, including a visual representation of the concept, its equivalent area distribution, or its overpressure signature, can also be quickly retrieved.

5.5 Multi-objective Interactive Optimization

A large amount of insight into the proof of concept problem was gained through the thorough exploration of the concept space described in the previous section. Unfortunately, no acceptable solution was found that also met all of the program goals established in Table 10. One remedy for this problem would be to add additional technologies that improve disciplinary metrics such as weight or specific fuel consumption, but this may dramatically increase the risk and cost associated with the

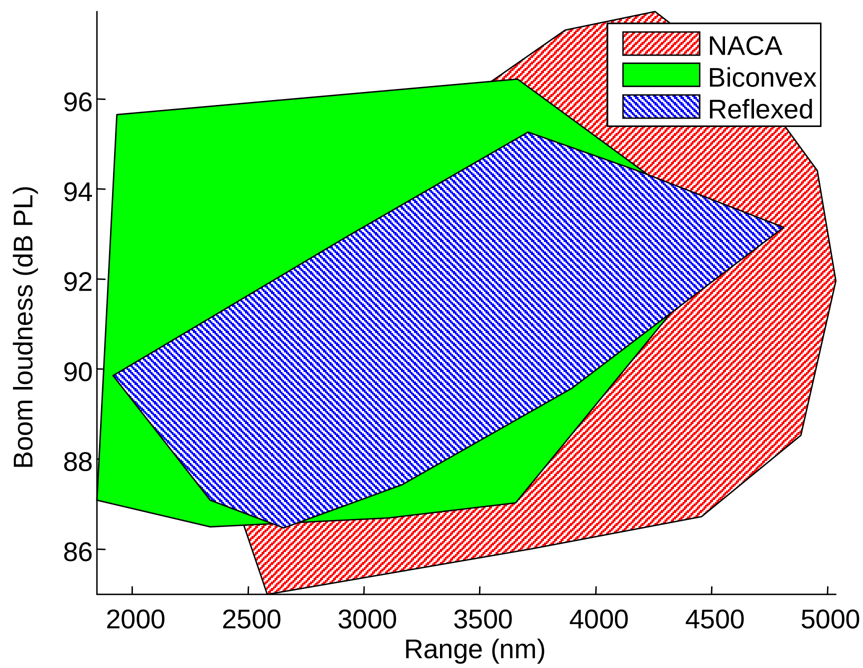


Figure 87: Impact of airfoil type on range and boom loudness

program. Instead, the results of the previous section were used to establish more realistic goals that should be attainable with current technology levels. These goals were used to form the value function that forms one of the objectives optimized by the Hybrid Interactive Genetic Algorithm.

In order to fully incorporate designer preference into the optimization procedure, a graphical user interface was created that displays the population members currently being evaluated and queries the engineer for a rank of how "good" each individual is (Figure 91). To ease the burden on the evaluator, only five subjective fitness values were allowed: Bad, Poor, OK, Good, and Best. Based upon the results of several usability tests, a "radio button" style input mechanism proved to be easier to operate than the "list box" form used in the past. In addition to subjectively evaluating

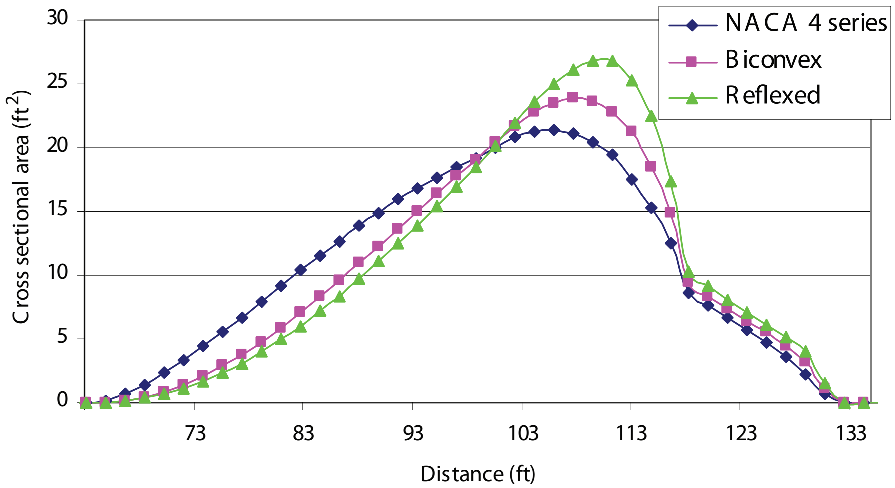


Figure 88: Impact of airfoil selection on longitudinal volume distribution of the wing

each design, the engineer is also allowed to update the goals and weights used to form the aggregate goal attainment metric, resulting in an IGA that is both "broad" and "narrow" according to Takagi's IGA classification system. [149] By clicking on the "edit" button, the user is also allowed to actively modify one of the concepts generated by the algorithm and re-insert it into the population.

The Hybrid Interactive Genetic Algorithm was implemented using the parameter settings given in Table 23 and the same computer cluster as was used to run the Multi-value Genetic Algorithm. After every four generations of interactive optimization of the primary population, the secondary population would be run using the K-Nearest Neighbor classifier as a surrogate for the human and the parameter settings listed in Table 23. Once this secondary population's termination criteria was met, the best designs from this population were presented to the expert for classification and injection into the primary population. During the execution of the algorithm, the

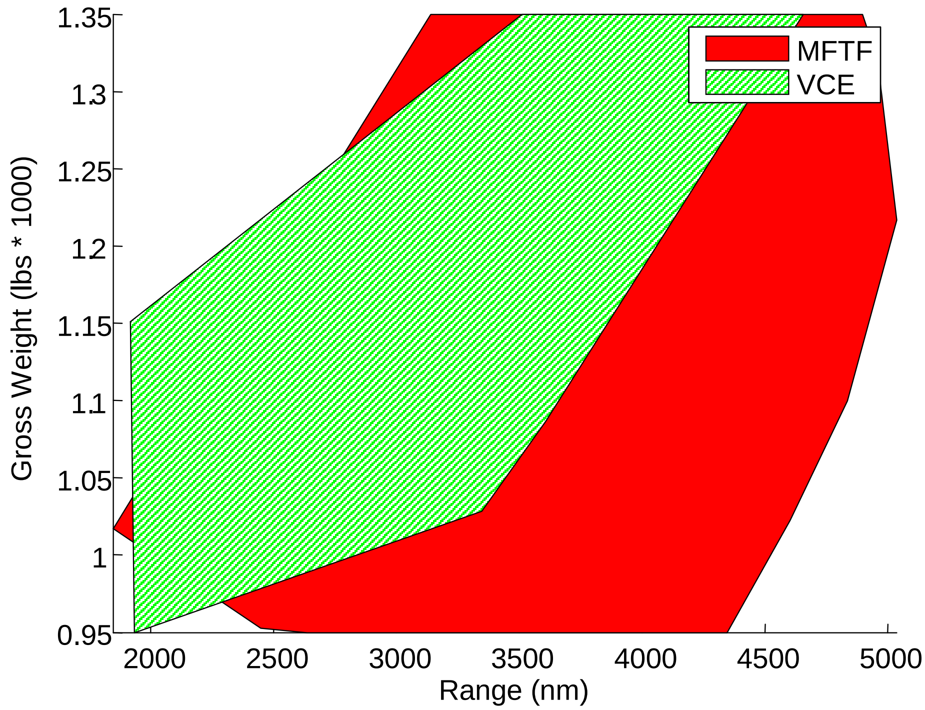


Figure 89: Impact of cycle selection on range and gross weight

user was allowed to vary the goals used to form the numerical value function via the GUI. The final values of these goals at the end of the optimization procedure are given in Table 24. Total execution time, including time allocated to human fitness assessment, was approximately two hours.

The results of the Proof of Concept problem are presented in Figure 92. As shown in the figure, the final population contained three Pareto-optimal solutions. Out of these designs, two have variable geometry wings and one has a fixed geometry "blended arrow" planform.

During the course of the interactive optimization procedure, the author noted that the algorithm was exploiting several aspects of the analytical model. First, the algorithm was frequently positioning the nacelles under the trailing edge of the

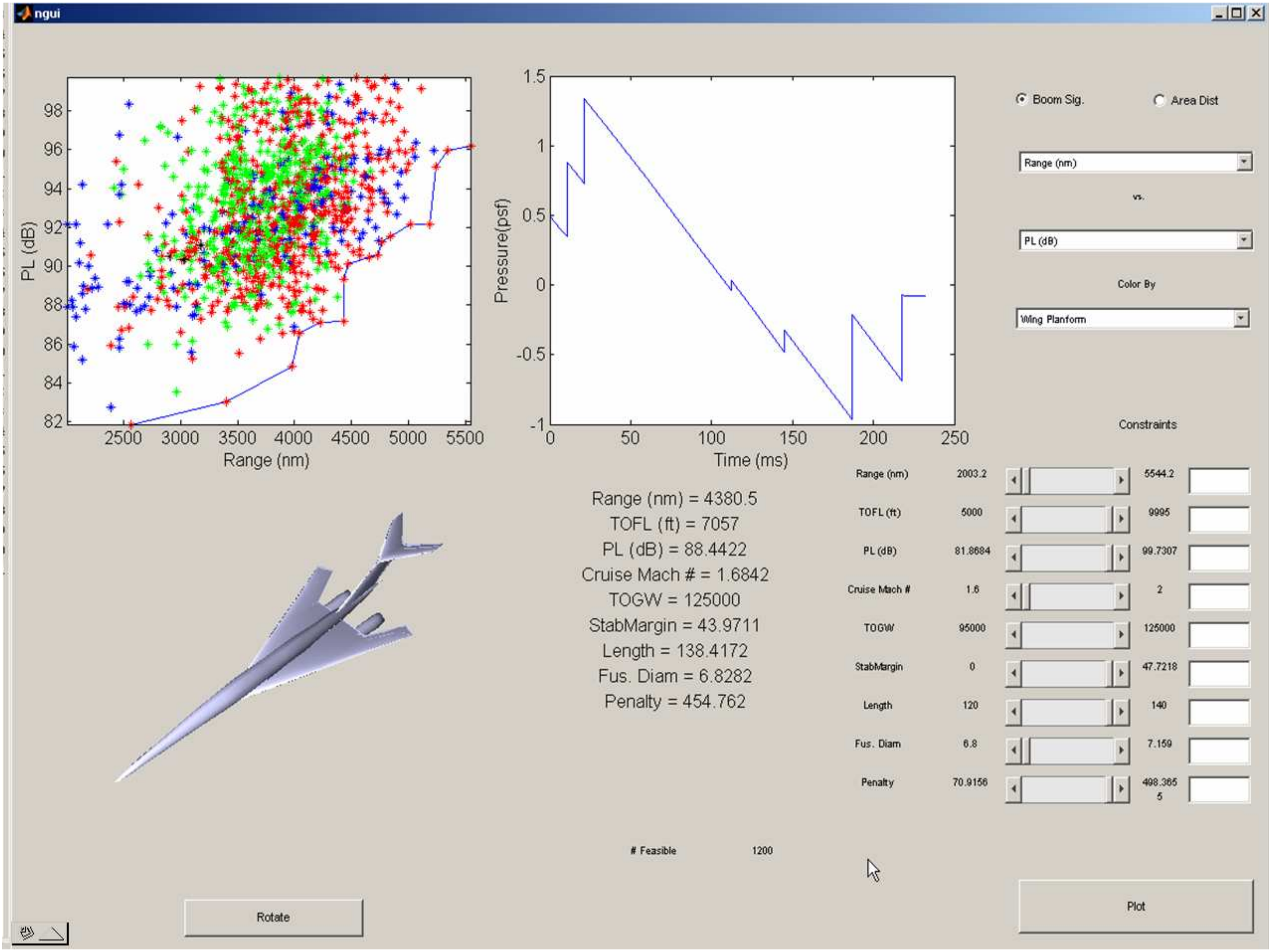


Figure 90: Graphical User Interface used to explore the results of the Multi-value Genetic Algorotihm

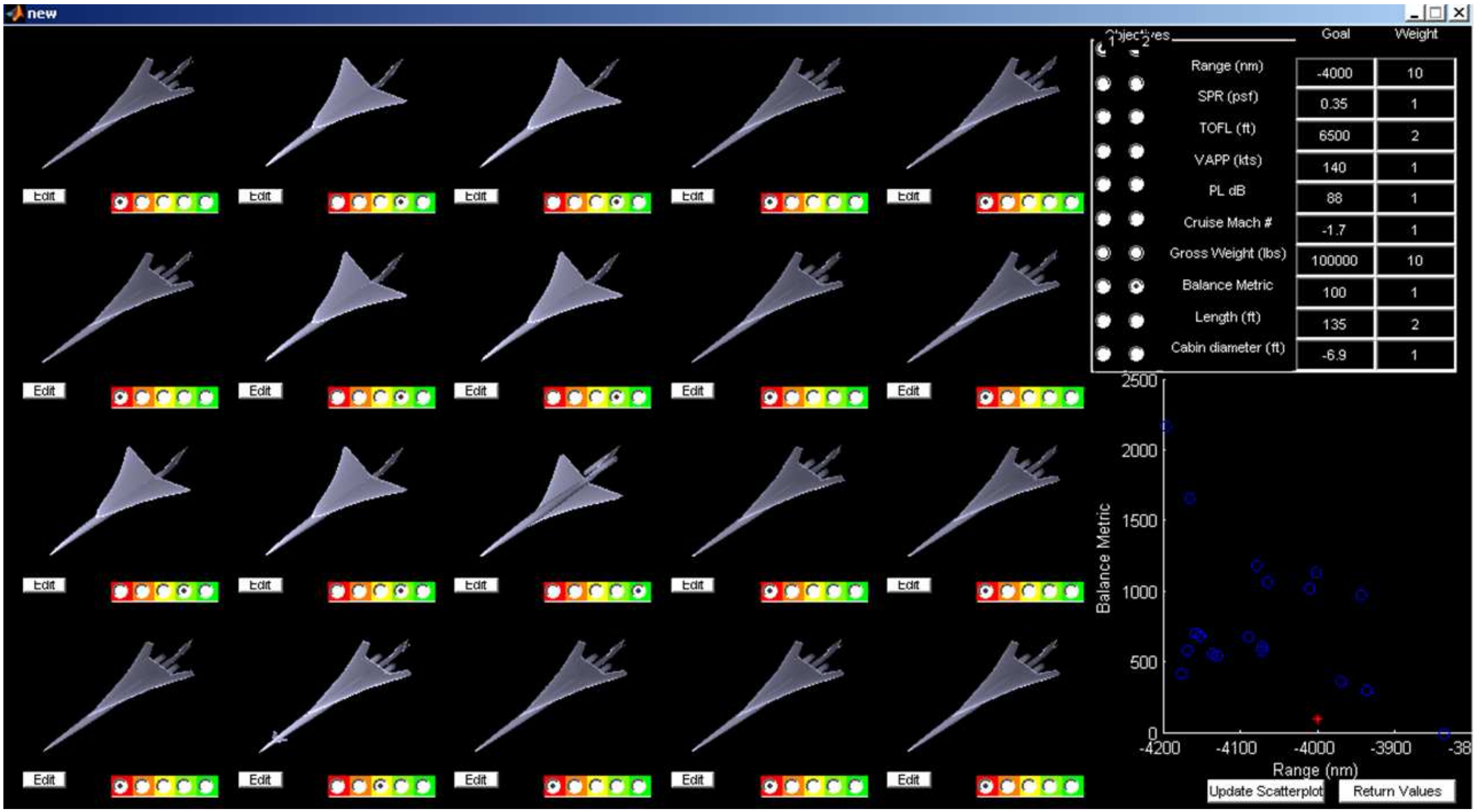


Figure 91: Proof of Concept Hybrid Interactive Genetic Algorithm interface

Table 23: Parameter settings for the interactive supersonic business jet optimization problem

	Primary population	Secondary population
Population size	20	200
Crossover rate	100%	70%
Crossover operator	Structured Crossover	(Algorithm 1)
Mutation rate (continuous)	2%	2%
Mutation rate (discrete)	2%	2%
Mutation type	uniform	uniform
Mating similarity parameter α	1	1
Mating similarity parameter β	3	3
RTR window size ω	1	200
Termination criteria	400 function evaluations	50 generations

Table 24: Objectives used by the HIGA value function

Objective	Value
Range (nm)	4200
Shock pressure rise (psf)	0.35
Takeoff field length (ft)	6500
Approach velocity (kts)	140
Boom loudness (dB PL)	88
Cruise Mach number (dB PL)	1.7
Gross weight (lb)	100,000
Balance metric	1
Length (ft)	140
Fuselage diameter (ft)	6.9

wing, which may lead to substantial flutter problems. [111] Although the analysis environment used for the proof of concept problem is not capable of capturing this phenomenon, the author accounted for it by penalizing the subjective preference rating of these concepts. Another issue encountered during the algorithm's execution was the tailless variable geometry concept. Although the algorithm was able to balance the aircraft as measured by the stability figure of merit, the author's engineering judgement suggests that this type of configuration is inadvisable.

It should be emphasized that by subjectively penalizing these designs, no information has been lost. They were able to survive the evolutionary process due to their superior performance as represented by the value function. In fact, the process has increased the amount of information available to the designer and fostered creativity. For example, the results of this procedure suggest that it may be worthwhile to perform detailed flutter analysis to determine if there would in fact be an issue with the aft-mounted nacelle configurations . Another interesting study would be to investigate the feasibility and possible benefits or drawbacks of including a retractable canard on the tailless variable geometry concept. In the end, it will be up to the decision maker which of these Pareto-optimal concepts should advance to the next stages of the design process.

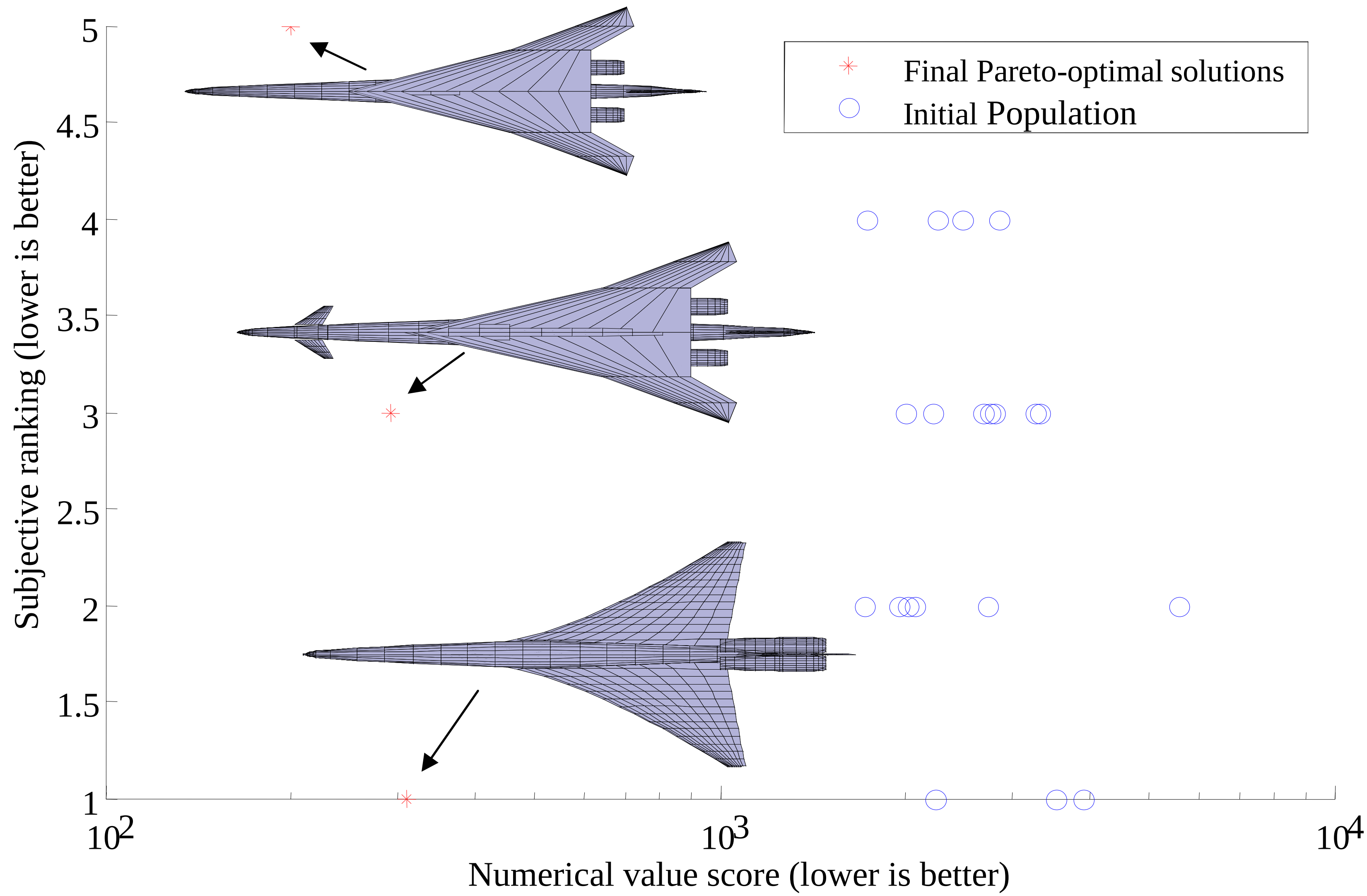


Figure 92: Pareto optimal solutions in the final IGA population

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

This dissertation has examined the difficulties associated with revolutionary system design and proposed a method based upon Evolutionary Computation designed to provide greater insight into the relationship between a vehicle's requirements and its promising attributes. By allowing the designer to provide feedback during the execution of the algorithm, the optimizer can be prevented from exploiting weaknesses in the analytical model without requiring the designer to over-constrain the problem. Additionally, the generation of a set of solutions enables the designer to observe the impact of his or her feedback on the outcome of the procedure. The feasibility of the method has been demonstrated using several simplified test cases and a supersonic business jet concept exploration example.

6.1 Research questions answered

Question 1: *How can engineering judgment and expertise be best combined with numerical analysis and optimization techniques to improve the conceptual design process?*

This thesis has demonstrated that by allowing the expert to interact with the optimization algorithm during its execution, it is possible to influence the outcome and ensure that usable solutions emerge from the process. This greatly increases the usefulness of optimization in the early stages in the design process because without user intervention, the optimization of conceptual design tools will often produce undesirable solutions.

It was found to be beneficial to actually let the designer interact in several ways. First among these is a mechanism that lets the designer examine and subjectively rank a design concept. For the present study this consisted of a visual examination of the solution, but other representations may be appropriate for other problems. Another way which the designer is allowed to is interact with the algorithm is by changing the goals and weights used by the numerical value function. Finally, an interface was created to let the designer modify solutions produced by the algorithm and re-inject these modified designs into the population to undergo further evolution.

Question 2: *Is it possible to rigorously search the Matrix of Alternatives for promising concepts without individually optimizing every possible alternative?*

Most methods for concept evaluation available in the literature require the analyst to individually optimize each alternative in order to perform a meaningful comparison. Because of this, designers are typically only able to perform an in-depth examination of a handful of design concepts. This thesis has demonstrated that through the application of a modified structured Genetic Algorithm, it is feasible to perform a thorough search of relatively large design hierarchies. Application of the method may force the engineer to consider solutions that would have previously been discarded outright, thereby encouraging creativity and “out of the box” thinking.

Question 3: *Is there a computationally feasible way to use physics-based analysis tools in addition to the historically-based design guidelines commonly used during the concept generation phase of routine design programs?*

The introduction of the Multi-value Genetic Algorithm has facilitated the optimization of complex analysis with several objectives. The explosion in computing capability over the past few years has also helped to make multiobjective optimization of relatively sophisticated analysis routines feasible: in the past three years alone, the amount of computational power available to the author has increased by three orders of magnitude. By investigating the non-dominated solutions that result from the optimization procedure, the engineer can gain valuable insight into the relationship between each of the requirements and into potential interactions between the system’s architecture and its requirements.

6.2 Revisiting the hypothesis

A method that allows designers to use physics-based analysis tools in concert with expert engineering judgement during the requirements and concept exploration stages of the design process will enable a more thorough examination of the combinatorial system alternatives matrix than is possible using traditional design practice. A method that meets these criteria can be obtained by combining the capabilities of Interactive and Multiobjective Evolutionary Computation, thereby facilitating the application of relatively sophisticated analysis methods to the task of concept exploration.

This work has successfully demonstrated that it is feasible to apply many of the analysis tools commonly used during the sizing phase of the design process earlier than they have been traditionally used. By leveraging this information sooner, the designer can increase his or her understanding of the problem and gain insight into the relationship between the requirements and favorable system alternatives. This application had not been possible using available design methods. Through the proof of concept problem, the union of Multiobjective and Interactive Evolutionary Computation has proven to be able to search relatively large design hierarchies and be effective at optimizing several objectives simultaneously. The results of this test problem revealed several interesting and non-obvious relationships that may help the engineer to avoid poor system selection choices and, in the end, produce a better concept.

6.3 Summary of Contributions

The objective of this research was the development of a new design method. In the course of this method's development several new capabilities have been developed that may prove to be useful even when used outside of the concept exploration method. These capabilities are now summarized.

The modified structured Genetic Algorithm: Through the use of the modified structured Genetic Algorithm introduced in Section 4.2, it is now possible to search relatively large and complex design hierarchies without individually optimizing every alternative. This enables the designer to examine many more alternatives than is feasible with traditional design methods, and hay help prevent the designer from choosing the "wrong" concept due to a lack of historical guidance or experience.

The Multi-value Genetic Algorithm: The Multi-value Genetic Algorithm introduced in Section 4.3.1.1 has been shown to be able to feasibly optimize several times the number of objectives possible using methods available in literature. By allowing the designer to treat more parameters as objectives rather than as constraints, the use of this algorithm may help to prevent the designer from over-constraining the problem before sufficient information has been gathered.

The hybrid Interactive Genetic Algorithm: The hybrid Interactive Genetic Algorithm described in Section 4.1.1 provides a mechanism for the design expert to provide feedback during the execution of the algorithm. Because this permits the designer to expand the domain searched by the algorithm, use of the method may lead to designs that exhibit performance better than those optimized using traditional methods with tight constraints. Although this algorithm has been demonstrated using a aircraft design example, it may be very useful in domains where the analyst must account for qualitative and quantitative criteria such as automobile or bridge design.

6.4 Future work and recommendations

The overall goal of this work has been to develop and demonstrate a conceptual design method capable of assisting in the decision-making process during the design of revolutionary vehicles. Although this goal has been accomplished, several issues remain to be addressed before the method reaches its full potential.

Incorporation of technology selection:

As mentioned in the introductory chapters of this work, technology selection and integration plays an extremely important role in any design effort. The concept exploration method developed in the course of this work was demonstrated using fixed technology assumptions, but this approach does not account for possibly significant interactions between the requirements, configuration, and technologies. Future efforts should investigate the feasibility of performing technology portfolio selection in parallel with concept exploration. This will require the introduction of a mechanism to prevent the algorithm from simply adding all possible technologies to the vehicle. One possible way to accomplish this would be to add minimization of technology investment or risk as an additional objective to be optimized by the Multi-value Genetic Algorithm.

Additional study into effective surrogates for human input:

This work used a K-Nearest Neighbor prediction model as a surrogate for the human decision maker to accelerate the convergence rate of the hybrid Interactive Genetic Algorithm. Although this model was demonstrated to be effective, it may be possible to more accurately predict human response by using more advanced techniques. One drawback of KNN interpolation is that the Euclidean distance used to measure similarity does not account for the relative importance of different variables. The use of a weighted KNN prediction scheme or Artificial Neural Networks may be able to account for this effect. One promising idea is to link with a House of Quality that is updated during the progression of the algorithm. The relative importance rankings assigned in the bottom of the House of Quality could be used as weights by the KNN interpolation algorithm, resulting in a more accurate estimate.

Investigation of more sophisticated analysis methods:

The analysis tools used for the proof of concept problem are largely based upon linear theory. Although the use of this type of analysis at the earliest stages of the concept exploration problem is appropriate and represents a significant advancement, it may be advantageous in the future to use more sophisticated analysis. The application of very high fidelity codes such as Navier Stokes or Euler analysis will likely remain overkill for the foreseeable future, but the author believes that integration of full potential or panel methods may be appropriate. This may be particularly important for supersonic aircraft, where non-linear phenomena such as differential area ruling and nacelle interference can have a strong impact on vehicle performance.

6.5 Concluding remarks

The design of revolutionary aircraft has always and will continue to be a challenging problem that requires ingenuity and creativity. This dissertation has introduced a method designed to encourage and augment this creativity by allowing the designer to explore many more concepts in more detail than is feasible using traditional design methods.

APPENDIX A

REVIEW OF SMALL SUPERSONIC TRANSPORT DESIGN STUDIES

One of the most important steps in the design of any product is the gathering of information from literature, patents, and other references. Although no small supersonic transport aircraft has ever entered production, members of industry, government and academia have conducted research into the feasibility of such a vehicle for more than forty years.[96] This appendix contains a brief review of early design studies, and a more in-depth discussion of several modern concepts.

A.1 Early design studies (1963-2000)

The first known small supersonic transport design study was conducted by students at the University of Colorado in 1963 (Figure 93). [96] The requirements for this vehicle were to fly 4 passengers 3500 nm at Mach 3. The design had a very cramped passenger compartment with four feet of cabin height, and assumed very aggressive structural and propulsion technologies, resulting in a gross weight of only 8400 pounds. [163]

A more realistic concept investigation was performed four years later in 1967 by students at Georgia Tech as a part of a senior design course. [44] The students actually investigated two configurations: one with a delta wing and another with an unswept trapezoidal planform similar to that of the F-104 Starfighter (Figure 94). In

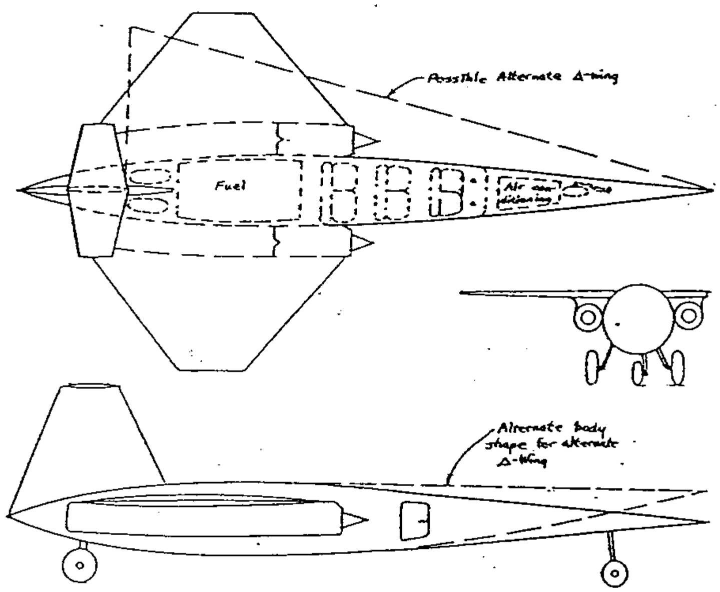
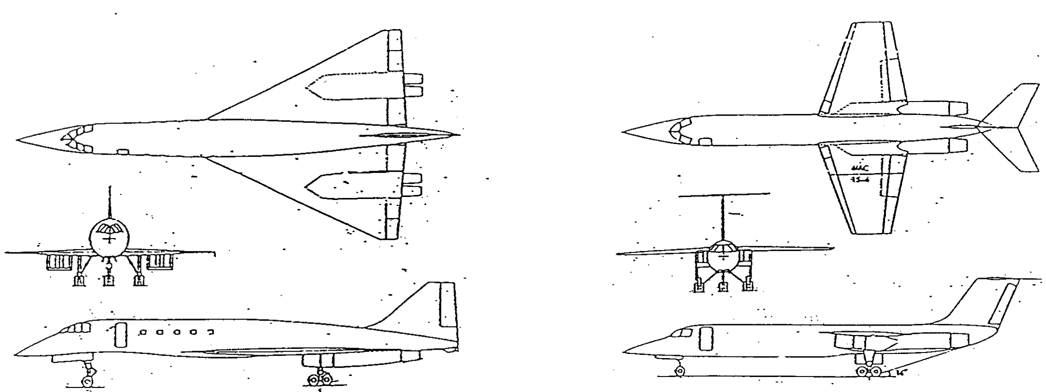


Figure 93: University of Colorado SBJ Design (1963) [163]

both cases the vehicle was sized to fly 10 passengers 3000nm at Mach 2.2, and operate out of 6000 ft runways. The passenger compartment was also significantly more realistic, with 6 feet of headroom. Although the students mentioned that environmental concerns including sonic boom, airport noise, and ozone impact would be important considerations, none of these factors was analyzed in depth.



(a) (b)

Figure 94: Georgia Tech (a) Delta and (b) Trapezoidal Wing Configurations (1967) [44]

From 1977 to the mid 1980s researchers at NASA Langley performed a large number of small supersonic transport design studies, beginning in-depth study of a number of arrow wing configurations with varying engine placement (Figure 95). [99] Each of the vehicles was sized to fly eight passengers over 3200 nm at Mach 2.2 and land within 6500 ft, and gross weight for the different concepts varied from 74,000 to 80,000 lbs. This research was the first to do detailed airport noise analysis using ANOPP [166] and concluded that it would be difficult to meet Stage III noise guidelines.

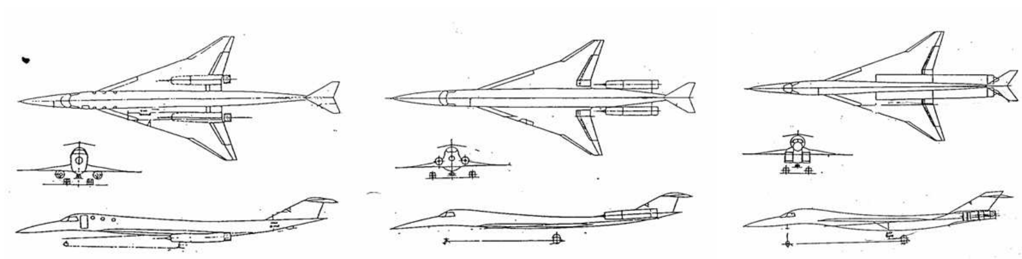


Figure 95: NASA Langley Supersonic Executive Jet Configurations (1977) [99]

As a follow-on to the Langely research in the early 1980s, Kentron investigated the impact of advanced technologies on the performance and weight of a number of vehicles with arrow and variable geometry planforms (Figure 96). [9] These vehicles had requirements very similar to those of the previous Langely studies, and the results indicated that the use of advanced materials would yield significant gross weight reductions. In each case, however, little emphasis was given to sonic boom or environmental effects.

In the late 1980's Sukhoi and Gulfstream both performed supersonic business jet feasibility studies, and in 1989 the two organizations began collaborating on a common

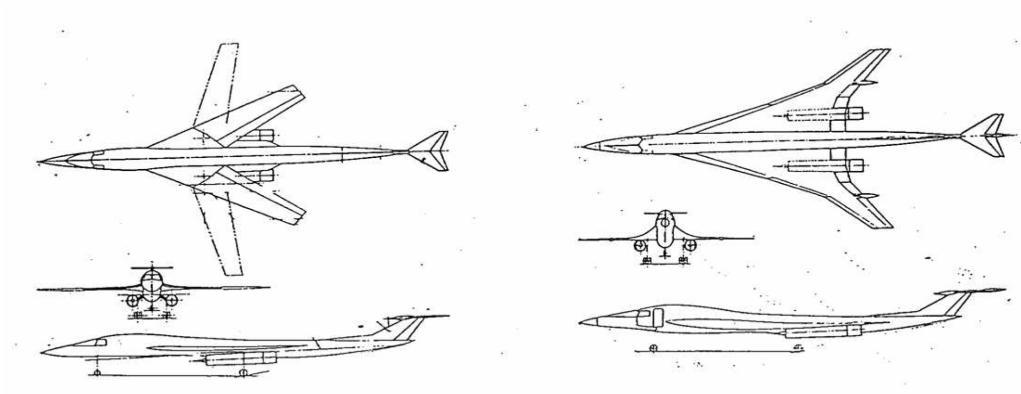


Figure 96: Kentron Supersonic Executive Jet Configurations (1984-86) [9]

design known as the S-21 (Figure 97). [41] The configuration had a takeoff gross weight of 100,000 lbs, and was designed to fly 10 passengers 4,400 nm at Mach 2.0 and operate from 6,500 ft runways. After several years of technical and market research, Gulfstream concluded that modern technology would not allow all requirements to be met and dropped out of the alliance. Despite Gulfstream's withdrawl, Sukhoi has continued to work on the S-21 design to the present day.

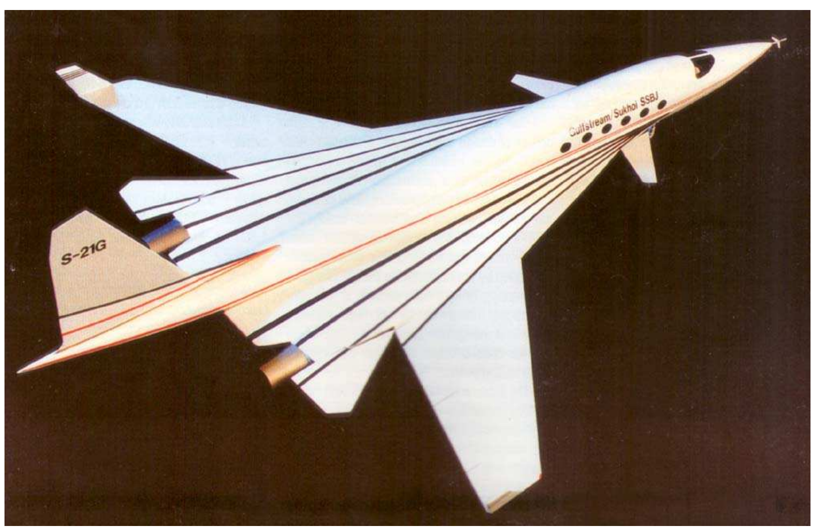


Figure 97: Sukhoi/Gulfstream S-21 Supersonic Business Jet (1989) [41]

In 1997 Dassault aviation unveiled plans for a supersonic business jet at the annual

NBAA convention. [55] Their design had a gross weight of 86,000 lbs and was designed to carry 8 passengers 4,500 nm at Mach 1.8 (Figure 98). Dassault elected to use a tri-jet configuration and a delta wing planform, and investigated propulsion systems based on the Snecma M88 or the General Electric F414. After a year of study, the organization concluded that there was no powerplant available that would provide adequate performance and have sufficient operating life, and dropped the project.

Figure 98: Dassault Supersonic Business Jet (1997) [55]

A.2 Modern design studies (2000-present)

The DARPA Quiet Supersonic Platform program of 2000 served to generate renewed interest in commercial supersonics. [160] The QSP program set extremely aggressive performance goals, with the objective of promoting advanced technology research (Table 25). It also placed much greater emphasis on sonic boom mitigation than previous studies, and funded a shaped boom demonstration program to demonstrate the validity of sonic boom theory. [121] After the program's conclusion in 2002, several airframe manufacturers continued work on conceptual design studies and have now released preliminary designs and requirements to the public.

Table 25: Quiet Supersonic Platform System Goals

Goal	Value
Gross Takeoff Weight	100,000 lbs
Cruise Mach Number	2.4
Cruise Range	6000 nm
Payload	20,000 lb
Noise	Stage III
Sonic Boom pressure rise	.3 psf

A.2.1 Aerion

Unlike all other modern supersonic business jet design concepts, the Aerion proposal is not designed for quiet supersonic flight. [112] The primary reason stated for this decision is risk mitigation, since FAR 91.187 currently prohibits supersonic flight over land, regardless of boom loudness. The fact that there is no low boom requirement greatly changes the design space, and allows the engineers to use a low sweep natural laminar flow planform to decrease viscous drag (Figure 99).

Another design decision that was influenced by the desire for low risk is the choice of powerplant. Rather than pursue a new supersonic engine design, Aerion engineers have decided to use a Pratt and Whitney JT8D-219, currently in service on the MD-80. Though this engine cycle was designed for subsonic operation, bother Aerion and P&W believe that with a new inlet, nozzle, and minor cycle modifications it could serve as an excellent propulsion system for their aircraft.

Figure 99: Aerion Supersonic Business Jet (2004) [112]

Apart from the fact that the design is not a low boom concept, other Aerion requirements seem in line with other industry players: range greater than 4000 nm, balanced field length of 6000 ft or less, ability to carry 8-12 passengers, compliance with stage IV noise regulations, and a 90,000 lb gross takeoff weight.

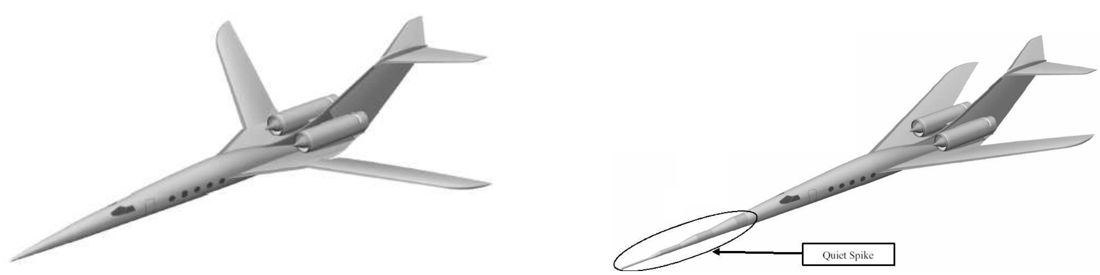
A.2.2 Gulfstream

As mentioned earlier, Gulfstream aerospace began studying small supersonic transports in the late 1980s, but after several years of research concluded that the technology available at the time was not advanced enough to allow all requirements to be met. In the late 1990s Gulfstream performed a number of market surveys and concluded that the potential market had increased substantially in the 1990s due to the increased popularity of fractional ownership, from 150 to 300 vehicles. [81]

Early configuration studies from the late 1990s investigated a fixed wing configuration with a T-tail and modified Rolls Royce Trent turbofan engines installed under the wings. Subsequent investigation led Gulfstream to reposition their design's nacelles and place them on the rear fuselage, where the engine's effect on the sonic boom signature would be minimized [71].

Other than the requirement for low boom, Gulfstream's top level requirements are rather similar to Aerion's: range between 4000 and 4800 nm, balanced field length of less than 6000 ft, 8-10 passengers, cruise Mach number of 1.6 to 1.8, better than Stage IV airport noise, and a gross weight of 100,000 lbs. [162]

After a thorough comparison of fixed and variable geometry planforms, Gulfstream engineers decided that a variable geometry wing's benefits outweigh its penalties, and the current configuration features such a planform (Figure 100(a)). [140] Another innovation featured in the most recent concept is an extendable nose spike that is deployed during supersonic cruise to dramatically decrease sonic boom loudness (Figure 100(b)). [72]



(a) (b)

Figure 100: Gulfstream variable geometry QSJ concept in (a) subsonic and (b) supersonic configuration
[161]

A.2.3 SAI/Lockheed

After a brief collaboration with Gulfstream between 1999 and 2000, Lockheed Martin began work on its own supersonic business jet design in 2001. This effort was actually funded by an entity known as Supersonic Aircraft Initiatives inc., which has invested approximately 25 million dollars to date. [112]

Like Gulfstream, SAI and Lockheed believe that boom mitigation is a requirement, and is targeting a 99% reduction in loudness compared to Concorde's boom. The SAI design is considerably heavier (153,000 lbs) and requires longer runways (8000 ft) than other proposed concepts. [112] Most of the major engine manufacturers are currently generating propulsion concepts for the vehicle, and it is expected that one of these will be selected as the powerplant within the next year. The most unique feature of the SAI concept is its tail-braced wing configuration, which provides greater stiffness and should help avoid some of the aeroelastic problems associated with rear mounted underwing engines (Figure 101). Although previous research has suggested that under-wing nacelles may have a detrimental effect on sonic boom loudness [71], Lockheed and SAI claim that this negative impact can be mitigated through careful wing reflexing [111].

Figure 101: SAI/Lockheed Supersonic Business Jet [112]

A.2.4 Northrop Grumman

Although it is known that Northrop Grumman is actively pursuing supersonic business jets, current information about the status of their design effort is difficult to come by. The most recent publication regarding their design work dates from 2002, and describes a concept designed to fulfill a hybrid of civil and military requirements inspired by the DARPA QSP goals. [87] These requirements are accordingly much more aggressive (and less realistic) than those found in more recent publications by other airframers.

The Northrop-Grumman design concept was developed in cooperation with Raytheon and features a highly swept joined wing with extensive natual laminar flow (Figure 102). It is also quite long, with a fuselage length of 156 ft and a total configuration length of nearly 170 ft. Given that several years have passed since the publication of these design studies, it is likely that Northrop Grumman is working on design alternatives with more modest requirements that are less reliant so many high risk technologies.

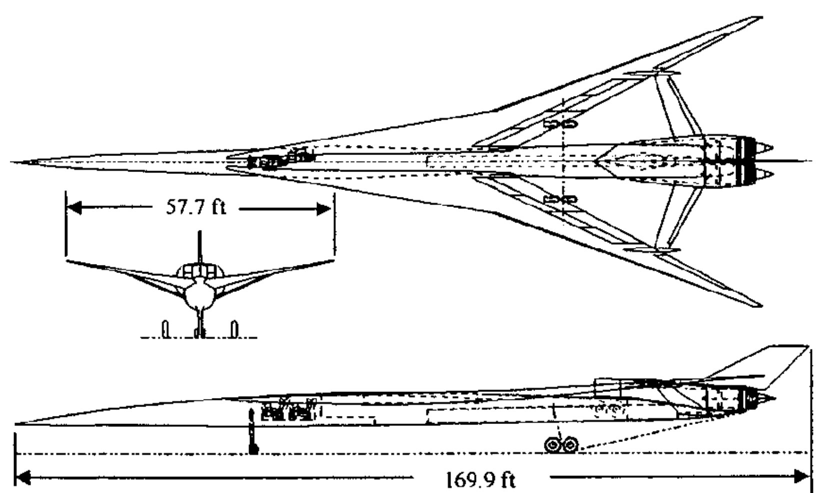
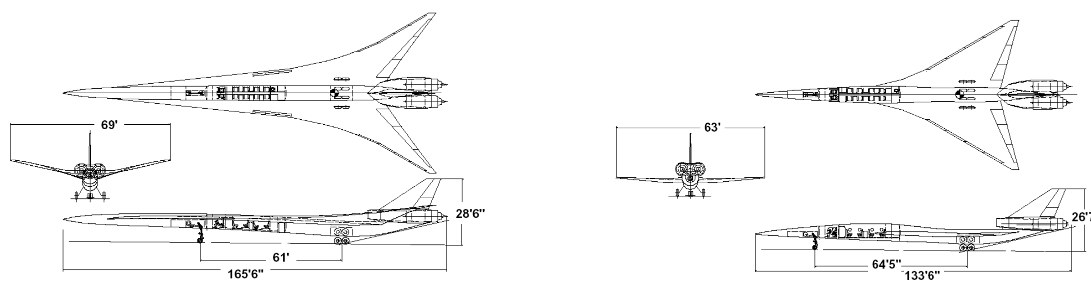


Figure 102: Northrop Grumman Supersonic Business Jet [87]

A.2.5 Raytheon

During the QSP program from 2000-2002, Raytheon worked as a subcontractor in support of Northrop-Grumman's design efforts. [118] Recent publications suggest that Raytheon is now working on new supersonic business jet concepts with requirements that are more realistic than those specified as a part of QSP. [5] [6] The most recent of these describes two configurations, one designed for unrestricted supersonic overland flight, and one with no shaped boom. The analysis showed that requiring low boom results in substantial penalties in other metrics: the low-boom design (Figure 103(a)) is 17% heavier and 30 feet longer than the equivalent high-boom design (Figure 103(b)).



(a) (b)

Figure 103: Raytheon supersonic business jets designed for (a) low boom and (b) high boom

[6]

APPENDIX B

SOURCE CODE FOR INTERACTIVE GENETIC ALGORITHM

```
%function ga
rand('state',sum(100*clock)); %seed the random number generator
%initialize variables
pswitch=1; %set 0 for standard function evaluation, 1 to use
... distributed computing psettings=[];
nmachines=19; % number of processors available
%RANGE SHIFT
rangeshift=0;
% RESTART
restart=0; restartinterval=4;
% LOCAL SEARCH % %these are methods like
localsearch=0; %use local search (hillclimb,sqp, or powell)
numsearches=30; %number of local searches to perform
surrogatesearch=0; %create a linear model from the DOE and use it to
... find promising solutions
surrogatetype='stepwise'; %options are 'linear', 'rs', 'stepwise', or 'kriging'
searchinterval=30; archiveinsert=0;
%run subproblems within the main GA
useMOsubproblem=1; subproblemint=5;
Subproblem.functionname='igasurrogatesearch';
Subproblem.numgen=50; Subproblem.popsize=100;
Subproblem.matingbeta=3; Subproblem.includedens=0;
% ************************************************************
% don't change these
model=[]; gensincerestart=0;
% REPLACEMENT %
replacement.switch = 'BF'; %options are 'PR' for parental replacement,
... 'TA' for threshold accepting, and 'BF' for best fitness
replacement.resettime=11; %number of generations before resetting ,
... used only with threshold accepting
replacement.coolingrate=.1; %selection pressure parameter for use
... with threshold accepting
replacement.removeduplicates=0; %allow multiple copies of a single
```

```
... solution to propagate?
% CROSSOVER SETTINGS
crossover='structured';%'structured'; %use the old crossover
... unless some of your factor settings are dependent on others,
... then use 'new'
crosstype='uniformlinear'; %options are '2pt' or 'uniform' for
... binary encoded problems, for real value problems additional
... options are '2ptlinear','uniformlinear'
pcross=.7; %percentage of time crossover occurs
matingrestriction=1; %for use with new crossover
matingbeta=0; %pos = more similar, neg = mroe dissimilar
disttype='decision';
% SELECTION SETTINGS
reproduction=2; %1 is roulette wheel, 2 is tournament selection
niching=0; %this setting is used for reproduction when
... you are using Pareto Optimality as a criteria
nicheradius=1/30; %this should be smaller the bigger your population gets
paretopreference=0; %directed pareto search
% MUTATION SETTINGS
mutation=2; %set to 1 for single point, 2 for field
muttype='uniform'; %options are 'bitflip' (for use with binary encoding),
... 'gaussian',or 'uniform' (these 2 for continuous
representations)
pmut=.05; %probability of mutation occurring ( per individual)
pfield=.2; %probability to use with field mutation (not used with single
point)
mutsigma=.1; %standard deviation to use with gaussian mutation
pstructmut=.02;;%mutation probability to use with configuration
... variables with the structured crossover
% ELITISM
elitism=0; %preserve the best individual from generation to generation -
... if used with MO problem the convex pareto set is retained from
gen to gen
% FITNESS FUNCTION
functionname='interactive';% 'runanal1' 'runanal1' 'runanal'};
fittype='MO';; %SO for single objective, MO for multi objective
% UTILITY SCALING
uscaling=0; mask=[1:11]; targetvals=1; targetnorms=1;
binsize=[];%[500 .3 500 1 .2 10000 20 5 .5 1 10];
constraints=[];%[-4000 .5 7000 88 -1.5 130000 10 140 -6 -1 500];
importance=ones(1,10);%ones(1,2);%[1 1];
MOtype='IGA'; includedens=0;
% POPULATION SETTINGS
popsize =20; %population size, usually set to around ~3 string length
numgen =100; %number of generations to run the GA for
```

```
% ISLAND GA SETTINGS
subpops = 1; %set this >1 if you want to have subpopulations
subgenerations =1;%how many generations in between migrations
... ( used if subpops > 1)
migrationrate = .5; %what percentage of individuals should migrate?
loadsaved=0; %load an old population (x) from saved.mat
savetomat=1; matfilename='matlab.mat';
%set up the problem
encoding='real'; %use either binary encoding or real encoding
numvar = 128;%67; %total number of variables required
... by objective function
numconfigvars=6;%5; %number of discrete options available
... - these variables should go first - set to 0 if using old
xover!!!!!
numcommonvars=35;%31; %these are variables like design
... mach number etc that are universal to all designs - ... should
be located after configvars
%these are used with new hierarchical crossover
categorytypes={[3 4 5 6] [0 1 2 3] [1 2 5] [0 1] [1] [1 2 4]};
... %these are the categorical options, for things like wing type --
... called here "config vars" numcategoryvars={[17 19 11 19] [0 6
6 6] [2 1 0] [0 0] [0] [0 0 0]};
... %these are the number of continuous variables
... associated with each "config var"
%these are the variable ranges (low and high bounds)
... - if using mixed categorical/continuous representations ...
non dimensional (ie 0-1) ranges are highly recommended
varrange=repmat([0 1],numvar-numconfigvars,1); if
isequal(encoding,'binary') bits=5 stringlength =
(numvar-numconfigvars)*bits+numconfigvars;
%find length of chromosome string
else bits=1; stringlength=numvar; end bestval = 10^10;
if loadsaved==0; % Select random initial population
if isequal(encoding,'binary') for p = 1:subpops
x(:,:,p)=randi(popsize,stringlength); end else for p = 1:subpops
x(:,:,p)=rand(popsize,stringlength); end end for i =
1:numconfigvars for p = 1:subpops ctype=categorytypes{i};
x(:,i,p)=ctype(randint(popsize,1,[1,length(ctype)])); end end
else %load saved population
load saved.mat end
% %*********** Initialize Distributed Computing ***********
switch pswitch case 1 addpath('../../mpi/MPI'); pdebug('on');
psetup('available',nmachines); psettings=[]; case 2
addpath('../codes/parmatlab')
[psettings.ss,psettings.rs]=initmajordomo; end
```

```
%****************************************************
% store settings to structure "Settings'
... that will be passed to the GA function
categorytypes={categorytypes}; numcategoryvars={numcategoryvars};
vars = eval('who')';
vals = eval(['{' sprintf('%s ', vars{:}) '}']);
c = [vars; vals]; Settings = struct(c{:});
%****************************************************
for s = 1:subgenerations for p = 1:subpops
[xout{p},outputvec{p},fitness{p},intout{p},trackovec{p}]= ...
runga(Settings(p),x(:,:,p)); save globalpop.mat end end
%
*******************************************************
************
% MAIN GA FUNCTION
%
*******************************************************
************ function
[x,outputvec,fitness,int,trackovec]=runga(Settings,xin,arg2)
mmv2struct(Settings); %unpack settings
fittype=Settings.fittype; %for some reason u gotta do this with matlab 6.5
if nargin == 2
x=xin; %use input x
[popsize,stringlength]=size(x); else
x=randi(popsize,stringlength); end if nargin == 3 archivex=xin;
archive=arg2; end if strcmp(replacement.switch,'eNSGA2') if nargin
< 3 archive=repmat(1000,1,sum(importance>0));
archivex=zeros(1,stringlength); end
Ba=repmat(1000,1,sum(importance>0));
[larchive,junk]=size(archive); end
executeindex=[1:popsize]'; %this is the ordered list of members
... of the population who need to have function evaluations
performed outputvec=[];compindex=[];outputvect=[];intt=[];
bestx=[];bestovec=[];bestcindex=[];
for m = 1:numgen % *************** BEGIN GENETIC
ALGORITHM ******************** if m == numgen
executeindex=[1:popsize]';
%the final generation we will run all cases just for checking
end
% ********************************* DECODE INPUT
*********************
int=decode(x,encoding,numconfigvars,bits,varrange);
% ********************************* PERFORM FUNCTION
EVALS ************ if strcmp(functionname,'interactive')
[outputvec,importance]=...
```

```
functioneval(functionname,executeindex,int,pswitch,psettings,...
outputvec,model,mask); else
[outputvec]=functioneval(functionname,executeindex,int,pswitch...
,psettings,outputvec,model,mask); end
% ************************************ STORE VALUES
******************
outputvect=[outputvect;outputvec(executeindex,:)];
intt=[intt;int(executeindex,:)]; if m == 1 origint=int;
origoutputvec=outputvec; end
% ************************************* FITNESS
TRANSFORMATION ******* if isequal(fittype,'MO') if uscaling
lnorm=max(outputvec); hnorm=min(outputvec); for i = 1:length(low)
vvv=[max(low(i)-(high(i)-low(i))/100,lnorm(i))... low(i) mid(i)
high(i) min(high(i)+(high(i)- low(i))/100,hnorm(i))];
soutputvec(:,i)=interp1(vvv,uuu,outputvec(:,i),'cubic'); end else
soutputvec=outputvec; end
fitness=MOfitness(soutputvec,soutputvec,importance,MOtype); else
fitness=outputvec; end
% ******************************** MODEL SEARCH (if
desired) ******* if surrogatesearch == 1 & m < numgen &
mod(m,searchinterval)==0
%use linear regression to model the system then perform
optimization on the linear model [modelx,modelovec]=...
modelsearch([origint;int],[origoutputvec(:,find(importance));...
outputvec(:,find(importance))],numvar,bits,varrange,functionname,...
pswitch,psettings,Settings);
modelint=decode(modelx,encoding,numconfigvars,bits,varrange);
%append so these results will be used to improve model deficiencies
next time origint=[origint;modelint];
origoutputvec=[origoutputvec;modelovec]; if
~strcmp(replacement.switch,'BF') [parlen,junk]=size(modelx);
[a,b]=sort(fitness); reproster=b(popsize:-1:(popsize-parlen+1));
x(reproster,:)=modelx; outputvec(reproster,:)=modelovec;
int(reproster,:)=modelint; end else if ( localsearch > 0 ) & m <
numgen & mod(m,searchinterval)==0 list=randperm(popsize);
%[a,list]=min(fitness);
switch localsearch
case 1 %use a hillclimber
[modelx,modelovec,modelfitness]=...
hillclimb(x(list(1:nmachines),:),...
outputvec(list(1:nmachines),:),...
fitness(list(1:nmachines),:),... bits,varrange,functionname,...
pswitch,psettings,numsearches,...
targetnorms,targetvals,fittype,M... Otype,encoding,numconfigvars);
case 2 %use SQP
```

```
[modelx,modelovec,modelfitness]=... sqp(x(list(1:nmachines),:),...
outputvec(list(1:nmachines),:),...
fitness(list(1:nmachines),:),...
bits,varrange,functionname,pswitch...
,psettings,numsearches,targetnorms,...
targetvals,fittype,MOtype,encoding,... numconfigvars);
case 3 %use powell's method search
for i = 1:popsize for j = 1:size(outputvec,2) w(i,j)=...
((max(outputvec(:,j))-outputvec(i,j))/...
(max(outputvec(:,j))-min(outputvec(:,j))))...
/(sum((max(outputvec)-outputvec(i,:))...
./(max(outputvec)-min(outputvec)),2)); end end
w=w./repmat(max(outputvec)- min(outputvec),length(outputvec),1);
[modelx,modelovec,modelfitness]... =powellsearch(x(list(1),:),...
outputvec(list(1),:),fitness(list(1),:)...
,bits,varrange,functionname,pswitch,...
psettings,numsearches,w(list(1),:),...
encoding,numconfigvars,mask); end
%insert
modelint=decode(modelx,encoding,numconfigvars,bits,varrange); if
~strcmp(replacement.switch,'BF') [parlen,junk]=size(modelx);
[a,b]=sort(fitness); reproster=b(popsize:-1:(popsize-parlen+1));
x(reproster,:)=modelx; outputvec(reproster,:)=modelovec;
fitness(reproster,:)=modelfitness; int(reproster,:)=modelint; end
else modelx=[]; modelovec=[]; end end
%% Subproblem search
if useMOsubproblem && mod(m,subproblemint)==0
i=1;% for i = 1:length(mask)
Gsettings=Settings; subproblemfieldnames=fieldnames(Subproblem);
for j = 1:length(subproblemfieldnames) eval(['Gsettings.'
subproblemfieldnames{j}... '=Subproblem.' subproblemfieldnames{j}
';']); end
% Gsettings.mask=mask(i);
% Gsettings.importance=1;
Gsettings.useMOsubproblem=0; Gsettings.importance=importance;
Gsettings.matfilename='matlab1.mat';
migx=rand(Gsettings.popsize,Gsettings.stringlength); for i =
1:numconfigvars ctype=categorytypes{i};
migx(:,i)=ctype(randint(Gsettings.popsize,1,[1,length(ctype)]));
end [subproblemx{i},subproblemoutputvec{i}]=runga(Gsettings,migx);
modelx=[]; for i = 1 modelx=[modelx;subproblemx{i}]; end if
size(modelx,1)>20 modelx=modelx(randint(20,1,[1
size(modelx,1)]),:); end save test.mat if
strcmp(functionname,'interactive') [modelovec,importance]=...
functioneval(functionname,1:size(modelx,1)...
```

```
,modelx,pswitch,psettings,outputvec,model,mask); else
[outputvec]=functioneval(functionname,1:size(modelx,1)...
,modelx,pswitch,psettings,outputvec,model,mask); end else
modelx=[]; modelovec=[]; end if archiveinsert & m > 1
[modelx]=run;
modelint=decode(modelx,encoding,numconfigvars,bits,varrange);
modelovec=functioneval(functionname,1,modelint,pswitch,...
psettings,outputvec,model,mask); end
% ****************************** ELITISM
****************************
if elitism & length(bestx) > 0 %elite insertion
disp('elitism') im=find(~ismember(bestx,x,'rows'));
%which members of bestx are NOT included in x currently
if length(im) > 0 [a,b]=sort(fitness);
int(b(popsize:-1:(popsize-length(im)+1)),:)=bestint(im,:);
fitness(b(popsize:-1:(popsize-length(im)+1)),:)=bestfit(im,:);
x(b(popsize:-1:(popsize-length(im)+1)),:)=bestx(im,:);
outputvec(b(popsize:-1:(popsize-length(im)+1)),:)=bestovec(im,:);
end end
% ****************************** REPLACEMENT
************************ if m > 1 |
strcmp(replacement.switch,'eNSGA2') switch replacement.switch case
'TA' [x,outputvec,fitness]=...
thresholdaccept(m,replacement.resettime,...
replacement.coolingrate,fitness,oldfitness,...
x,xold,outputvec,outputvecold); case 'eNSGA2'
% UPDATE ARCHIVE
minpop=min(outputvec(:,find(binsize)));
maxpop=max(outputvec(:,find(binsize)));
outputvec(isnan(outputvec(:,1)),:)=...
repmat(max(outputvec),sum(isnan(outputvec(:,1))),1);
Bf=floor((outputvec(:,find(binsize))-...
repmat(minpop,popsize,1))./repmat(binsize(find(binsize))...
,popsize,1)); if m > 1 minpop=min(archive);
Bf=floor((outputvec(:,find(binsize))-...
repmat(minpop,popsize,1))./...
repmat(binsize(find(binsize)),popsize,1));
[larchive,junk]=size(archive); Ba=floor(((archive)-...
repmat(minpop,larchive,1))./...
repmat(binsize(find(binsize)),larchive,1)) end
fitness=MOfitness(outputvec,targetnorms,targetvals,'GOALSEEK');
for i = 1:popsize checkdom=sum(Ba<=repmat(Bf(i,:),larchive,1),2);
if any(checkdom==junk) & ~ismember(Bf(i,:),Ba,'rows')
%if the new member is dominated, reject it
else if any(checkdom==0)
```

```
%if the new member dominates an existing
... archive member, delete the existing and add the new
archive(checkdom==0,:)=[]; archivex(checkdom==0,:)=[];
Ba(checkdom==0,:)=[];
archive=[archive;outputvec(i,find(binsize))];
archivex=[archivex;x(i,:)]; Ba=[Ba;Bf(i,:)];
else if ismember(Bf(i,:),Ba,'rows') %its a member of the same
hyper-box junk2=find(ismember(Bf(i,:),Ba,'rows')); if
sum(outputvec(i,find(binsize))<archive(junk2(1),:))==0 %it normally
dominates, accept it
archive(junk2(1),:)=outputvec(i,find(binsize));
archivex(junk2(1),:)=x(i,:); else df=((outputvec(i,find(binsize))-
Bf(i,:))./binsize(find(binsize))).^2; da=((archive(junk2(1),:)-
Bf(i,:))./binsize(find(binsize))).^2; if df < da
archive(junk2(1),:)=outputvec(i,find(binsize));
archivex(junk2(1),:)=x(i,:); end end else
archive=[archive;outputvec(i,find(binsize))];
archivex=[archivex;x(i,:)]; Ba=[Ba;Bf(i,:)]; end end end
[larchive,junk]=size(archive); end if m > 1 for i = 1:popsize if
all(outputvec(i,:)>=outputvecold(i,:)) x(i,:)=xold(i,:);
outputvec(i,:)=outputvecold(i,:); end end end
%end
case 'BF'
%this should really only be used with pareto optimality i think...
pops = [xold;x;modelx];
[pops,unique2index]=unique(pops,'rows'); %make unique
ovec=[outputvecold;outputvec;modelovec];
ovec=ovec(unique2index,:); if uscaling lnorm=max(ovec);
hnorm=min(ovec); clear sovec for i = 1:length(low)
vvv=[max(low(i)-(high(i)-low(i))/100,lnorm(i)) low(i) mid(i)
high(i) min(high(i)+(high(i)-low(i))/100,hnorm(i))];
sovec(:,i)=interp1(vvv,uuu,ovec(:,i),'cubic'); end else
sovec=ovec; end if includedens tov=[sovec
crowded(categorytypes,intt,pops,ovec)]; if
~isequal(MOtype,'cSPEA2') fit=MOfitness(tov,tov,[importance
1],MOtype);%(max(tov(:,end))-min(tov(:,end)))/20],MOtype);
else fit=MOfitness(tov,tov,[importance min(tov(:,end))/2],MOtype);
end else fit=MOfitness(sovec,sovec,importance,MOtype); end
[fit,si]=sort(fit); pops=pops(si,:); ovec=ovec(si,:);
sovec=sovec(si,:); if isequal(MOtype,'SPEA2') |
isequal(MOtype,'eSPEA2') | isequal(MOtype,'cSPEA2')
numberofnondominated=sum(fit<1);
if numberofnondominated > popsize %need to truncate the list
maximum=max(sovec(1:numberofnondominated,find(importance)));
[minimum,jjjj]=min(sovec(1:numberofnondominated,find(importance)));
```

```
normalized=(sovec(1:numberofnondominated,find(importance))-
repmat(minimum,numberofnondominated,1))./(repmat(maximumminimum,
num berofnondominated,1)); [a,b]=min(normalized);
dist=distmat1(normalized);%.*distmat1(pops);
%%%%% temp
grossdistance=[]; for j = 1:size(numberofnondominated,1)
grossdistance(j,:)=sum((repmat(pops(j,1:numconfigvars),numberofnondomi
nated,1)~=pops(1:numberofnondominated,1:numconfigvars)),2 ); end
gd=distmat1(pops(1:numberofnondominated,(numconfigvars+1):end));
gd=(gd-min(gd))./std(gd);
dist=grossdistance+gd; %the smaller the gross distance, the
more genotypically similar the solutions grossdistance=[];
% ****************************
dist(:,b)=100; dist(b,:)=100; includedguys=1:numberofnondominated;
deletedguys=[];
[sorteddist,sortindex]=sort(dist(includedguys,includedguys));
% i think something funny is going on here ...
%sorteddist(ismember(sortindex,b))=100;
%sorteddist(:,b)=100;
% ***************************************
for i = 1:(numberofnondominated-popsize)
mindist1=min(sorteddist(2,:));
maybe=find(sorteddist(2,:)==mindist1);
[a,b]=max(sorteddist(3,maybe)); badguy=sortindex(2,maybe(b));
sorteddist(:,maybe(b))=[]; sortindex(:,maybe(b))=[];
sorteddist(sortindex==badguy)=[]; sortindex(sortindex==badguy)=[];
sorteddist=reshape(sorteddist,numberofnondominatedi,
numberofnondominated-i);
sortindex=reshape(sortindex,numberofnondominatedi,
numberofnondominated-i); deletedguys=[deletedguys;badguy]; end
pops(deletedguys,:)=[];ovec(deletedguys,:)=[];
fit(deletedguys,:)=[];sovec(deletedguys,:)=[]; end end
x=pops(1:popsize,:); outputvec=ovec(1:popsize,:);
fitness=fit(1:popsize,:);
int=decode(x,encoding,numconfigvars,bits,varrange);
sovec=sovec(1:popsize,:); case 'PR'
%do nothing (replacement has automatically occured)
end end
% ************** results tracking,recording and elite saving
**********************
%elite saving
if elitism if isequal(MOtype,'Pareto') & isequal(fittype,'MO')
eliteindex=paretoconvex(outputvec(:,find(importance)),'points');
bestfit=fitness(eliteindex,:); bestint=int(eliteindex,:);
bestx=x(eliteindex,:); bestovec=outputvec(eliteindex,:); else
```

```
[bestfit,fitindex]=min(fitness); bestint=int(fitindex,:);
bestx=x(fitindex,:); bestovec=outputvec(fitindex,:); end end
xold=x;oldfitness=fitness;outputvecold=outputvec;intold=int;
%performance tracking values
bestval=min(bestval,min(fitness)); trackvalint(m,:) =
min(fitness); trackval(m,:) = bestval; trackmean(m,:) =
mean(fitness); trackovec(m,:)=min(outputvec);
trackbestx(:,:,m)=bestx; disp('selection');
% ******************************* SELECTION
*************************** if niching == 1
ncount=nichecount(outputvec(:,find(importance)),nicheradius); else
if paretopreference == 1
ncount=MOfitness(outputvec(:,find(importance)),outputvec(:,find(importanc
e)),importance(find(importance)),'SWGR'); ncount=ncount'; else
ncount=[]; end end
%
scatter(outputvec(fitness<1,1),outputvec(fitness<1,4),20,x(fitness<1,1),'fille
d')
% axis([-7000 -2000 82 105])
% drawnow
% hold on;
% NN(m)=getframe;
if savetomat save(matfilename); end
if m == numgen %is the GA done running?
save; if strcmp(replacement.switch,'eNSGA2') x=archivex;
outputvec=archive; end
return %exit function
end
%%%%%%%%%%%%%%%%%% REPRODUCTION
%%%%%%%%%%%%%%%%%%%%%%
if strcmp(replacement.switch,'eNSGA2')
%randon replacement from archive - kinda sucky i think
%temp9=randint(1,popsize,[1 larchive]);;
%cross(2:2:popsize,:)=archivex(temp9(1:length(2:2:popsize)),:);
archivefitness=sum(archive>repmat(constraints,size(archive,1),1),2);
awinnerindex=tournament(archivefitness,ncount); if (popsize+1)/2 <
larchive temp9=awinnerindex(1:length(2:2:popsize)); else
temp9=randi(length(2:2:popsize),1,[1 larchive]); end
%select half of parents from archive
%cross(2:2:popsize,:)=archivex(temp9,:);
%select other half based upon similarity to those in archive, and
%tournament
ncomparisons=3; tmatrix=[tournament(fitness,[])'
tournament(fitness,[])' tournament(fitness,[])'];
%normalize
```

```
maximum=max(outputvec(:,find(importance)));
[minimum,jjjj]=min(outputvec(:,find(importance)));
normalized=(outputvec(:,find(importance))-
repmat(minimum,popsize,1))./(repmat(maximum-minimum,popsize,1));
narchive=(archive(:,find(importance))-
repmat(minimum,size(archive,1),1))./(repmat(maximumminimum,
size(archive,1),1));
%truncate
tmatrix=tmatrix(1:popsize/2,:); for i = 1:popsize/2
dist(i,:)=distmat1(narchive(temp9(i),:),normalized(tmatrix(i,:),:));
end [a,b]=min(dist,[],2); for i = 1:length(b)
pwin(i)=tmatrix(i,b(i)); end
%re assign values based on results of selection
x1(2:2:popsize,:)=archivex(temp9,:); x1(1:2:popsize,:)=x(pwin,:);
outputvec(1:2:popsize,:)=outputvec(pwin,:);
outputvec(2:2:popsize,:)=archive(temp9,:); else switch
reproduction
case 1 %%%%%%%%%%ROULETTE SELECTION
%%%%%%%%%%%%%%%
winnerindex=roulette(fitness,ncount); %returns roster of winners
case 2 %%%%%%%%%%TOURNAMENT SEL
%%%%%%%%%%%%%%%
winnerindex=tournament(fitness,ncount); %returns roster of
winners end if matingrestriction&matingbeta~=0 switch disttype
case 'decision' for j = 1:popsize
grossdistance(j,:)=sum((repmat(x(j,1:numconfigvars),popsize,1)~=x(:,1:num
configvars)),2); end gd=distmat1(x(:,(numconfigvars+1):end));
gd=(gdrepmat( min(gd,[],2),1,size(gd,2)))./repmat(max(gd,[],2)-
min(gd,[],2),1,size(gd,2));
grossdistance=grossdistance+gd; %the smaller the gross
distance, the more genotypically similar the solutions case
'objective' maximum=max(outputvec(:,find(importance)));
[minimum,jjjj]=min(outputvec(:,find(importance)));
normalized=(outputvec(:,find(importance))-
repmat(minimum,popsize,1))./(repmat(maximum-minimum,popsize,1));
grossdistance=distmat1(normalized); end
grossdistance=grossdistance*sign(matingbeta); %if matingbeta is
negative, we want dissimilar parents (promote diversity)
[ajunk,bjunk]=sort(grossdistance);
%run matingbeta tournaments, and pick the winner that is closest
genotypically to the solutions already picked tmatrix=[]; for kkk
= 1 :abs(matingbeta); tmatrix=[tmatrix tournament(fitness,[])'];
end for j=2:2:popsize
[jjj,yi]=sort(grossdistance(winnerindex(j-1),tmatrix(j,:))); if
tmatrix(j,yi(1)) ~= winnerindex(j-1) ind=1; else ind=2; end
```

```
winnerindex(j)=tmatrix(j,yi(ind)); end end
%re assign values based on results of selection
winner=int(winnerindex,:); x1=x(winnerindex,:);
fitness=fitness(winnerindex); outputvec=outputvec(winnerindex,:);
end disp('crossover');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% CROSSOVER
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%
cross=x1;
executeindex=[]; %as of right now, we know the fitness of every
individual.. switch crossover
case 'old' %% canonical crossover %%
postcross=cross; for j = 1:2:popsize if rand < pcross
executeindex=[executeindex;j;j+1]; %these designs must be reevaluated
[postcross(j,1:stringlength),postcross(j+1,1:stringlength)]=crossvec([postcro
ss(j,1:stringlength)],[postcross(j+1,1:stringlengt h)],crosstype);
end end
case 'structured' %% canonical crossover %%
postcross=cross; for j = 1:2:popsize if rand < pcross
executeindex=[executeindex;j;j+1]; %these designs must be reevaluated
[postcross(j,1:numconfigvars),postcross(j+1,1:numconfigvars)]=crossvec([p
ostcross(j,1:numconfigvars)],[postcross(j+1,1:numconfi
gvars)],'uniform');
[postcross(j,(numconfigvars+1):stringlength),postcross(j+1,(numconfigvars+
1):stringlength)]=crossvec([postcross(j,(numconfigvar
s+1):stringlength)],[postcross(j+1,(numconfigvars+1):stringlength)],crosstyp
e); end end
case 'structured2' % this crossover is like object oriented, but with
redundant genetic material postcross=cross; for j = 1:2:popsize if
rand < pcross
executeindex=[executeindex;j;j+1]; %these designs must be reevaluated
% -- SWAP COMPONENTS DIRECTLY IF THEY ARE OF
DISSIMILAR TYPE (with some probability), ELSE
%PERFORM CROSSOVER INDIVIDUALLY ON EACH
SIMILAR
%COMPONENT
crossindex=round(rand(1,numconfigvars)); %which of the
configvars will be swapped if dissimilar
crossmask=cmask(postcross(j:(j+1),(numconfigvars+1):stringlength),categor
ytypes,numcategoryvars,numconfigvars,numcommonvars,cro
ssindex); % vales with 1 undergo normal crossover, -1s swap, and 0s dont
change swapmask=[logical(crossindex) crossmask==1];
crossmask=[logical(zeros(1,numconfigvars)) crossmask==-1];
[postcross(j,:),postcross(j+1,:)]=swapvals(postcross(j,:),postcross(j+1,:),swa
```

```
pmask);
[postcross(j,crossmask),postcross(j+1,crossmask)]=crossvec([postcross(j,cro
ssmask)],[postcross(j+1,crossmask)],crosstype); end end
case 'new' % %%%% NEW OBJECT ORIENTED CROSSOVER
%%%
postcross=cross; for i = 1:popsize
compindex(i,1)=numcommonvars+numconfigvars+1; for j =
2:(length(categorytypes)+1)
compindex(i,j)=compindex(i,j-1)+bits*(numcategoryvars{j-
1}(find(categorytypes{j-1}==postcross(i,j-1)))); end end for j =
1:2:popsize
if rand < pcross %does crossover
happen?? executeindex=[executeindex;j;j+1];
%STEP 1 -- COMPARE GROSS CONFIGURATION OF
PARENTS
begincomps=1+numconfigvars+numcommonvars*bits; %where
does information about the first component begin
comparevector=cross(j,1:numconfigvars)==cross(j+1,1:numconfigvars);
%Figure out what components are common
and figure out where each individual component begins
crossindex(1:numconfigvars)=ones(1,numconfigvars); %all sub
components are available for crossover
%STEP 2 -- SWAP COMPONENTS DIRECTLY IF THEY
ARE OF DISSIMILAR TYPE (with some probability), ELSE
%PERFORM CROSSOVER INDIVIDUALLY ON EACH
SIMILAR COMPONENT
%CROSSOVER ON THE COMMON VARS
[postcross(j,numconfigvars+1:begincomps-
1),postcross(j+1,numconfigvars+1:begincomps-
1)]=crossvec(cross(j,numconfigvars+1:beginc
omps-1),cross(j+1,numconfigvars+1:begincomps-1),crosstype);
ii1=begincomps; ii2=begincomps; for kk = 1:numconfigvars
if comparevector(kk) == 0 & crossindex(kk) == 1 %the
components ARE different
if rand < .5 %swap the components
postcross1=cross(j+1,compindex(j+1,kk):compindex(j+1,kk+1)-1);
postcross2=cross(j,compindex(j,kk):compindex(j,kk+1)- 1);
postcross(j,ii1:ii1+compindex(j+1,kk+1)-1-
compindex(j+1,kk))=postcross1;ii1=ii1+compindex(j+1,kk+1)-1-
compindex(j+1,kk); postcross(j+1,ii2:ii2+compindex(j,kk+1)-1-
compindex(j,kk))=postcross2;ii2=ii2+compindex(j,kk+1)-1-
compindex(j,kk); postcross(j,kk)=cross(j+1,kk);
postcross(j+1,kk)=cross(j,kk);
else %retain the current component
postcross1=cross(j,compindex(j,kk):compindex(j,kk+1)- 1);
```

```
postcross2=cross(j+1,compindex(j+1,kk):compindex(j+1,kk+1)-1);
postcross(j,ii1:ii1+compindex(j,kk+1)-1-
compindex(j,kk))=postcross1;ii1=ii1+compindex(j+1,kk+1)-
compindex(j+1,kk); postcross(j+1,ii2:ii2+compindex(j+1,kk+1)-1-
compindex(j+1,kk))=postcross2;ii2=ii2+compindex(j,kk+1)-
compindex(j,kk); end end
if comparevector(kk) == 1 & crossindex(kk) == 1 %the
components are of the same type, breed them using standard
crossover
[comp1,comp2]=crossvec(cross(j,compindex(j,kk):compindex(j,kk+1)-
1),cross(j+1,compindex(j+1,kk):compindex(j+1,kk+1)-1),crosstyp e);
postcross1=comp1; postcross2=comp2;
postcross(j,ii1:ii1+compindex(j+1,kk+1)-1-
compindex(j+1,kk))=postcross1;ii1=ii1+compindex(j+1,kk+1)-1-
compindex(j+1,kk); postcross(j+1,ii2:ii2+compindex(j,kk+1)-1-
compindex(j,kk))=postcross2;ii2=ii2+compindex(j,kk+1)-1-
compindex(j,kk); end
if crossindex(kk) == 0 %reuse the old component
postcross1=cross(j,compindex(j,kk):compindex(j,kk+1)-1);
postcross2=cross(j+1,compindex(j+1,kk):compindex(j+1,kk+1)-1);
postcross(j,ii1:ii1+compindex(j,kk+1)-1-
compindex(j,kk))=postcross1;ii1=ii1+compindex(j+1,kk+1)-
compindex(j+1,kk); postcross(j+1,ii2:ii2+compindex(j+1,kk+1)-1-
compindex(j+1,kk))=postcross2;ii2=ii2+compindex(j,kk+1)-
compindex(j,kk); end end end end end disp('mutation');
%mutation
for j = 1:popsize if rand < pmut executeindex=[executeindex;j];
switch mutation case 1 u =
randi(1,1,[numconfigvars+1,stringlength]);
%random number to find mutation location
case 2
u=randi(1,round(pfield*stringlength),[numconfigvars+1,stringlength]);
end switch muttype case 'bitflip' postcross(j,u) =
~postcross(j,u); case 'uniform' postcross(j,u)=rand(1,length(u));
case 'gaussian' postcross(j,u)=normrnd(postcross(j,u),mutsigma);
end end end
%structured mutation
for i = 1:numconfigvars which=find(rand(1,popsize)<pstructmut);
ctype=categorytypes{i};
postcross(which,i)=ctype(randint(length(which),1,[1,length(ctype)]));
end
%record old values
x=postcross;
% range shift
if rangeshift newrange=prctile(int,[5 95])';
```

```
delta=newrange(:,2)-newrange(:,1);
newrange(:,1)=newrange(:,1)-delta/10;
newrange(:,2)=newrange(:,2)+delta/10;
newrange(:,1)=max(0,newrange(:,1));
newrange(:,2)=min(1,newrange(:,2));
int=decode(x,encoding,numconfigvars,bits,varrange);
varrange=newrange;
x=encode(int,encoding,numconfigvars,bits,varrange); end
%truncate decision variables to ensure we stay inside the bounds
x(:,(numconfigvars+1):stringlength)=max(x(:,(numconfigvars+1):stringlengt
h),0);
x(:,(numconfigvars+1):stringlength)=min(x(:,(numconfigvars+1):stringlengt
h),1);
if replacement.removeduplicates == 1 %prevent one individual from
taking over the population
[a,uniqueindex]=unique(x,'rows'); %which members are unique
nonunique=setdiff([1:popsize]',uniqueindex); %which members are not
%mutate all non unique members
for j = nonunique executeindex=[executeindex;j]; for k =
(numconfigvars+1):stringlength if rand < .1 switch muttype case
'bitflip' x(j,k) = ~postcross(j,k); case 'uniform' x(j,k)=rand;
case 'gaussian' x(j,k)=normrnd(x(j,k),mutsigma); end end end end
end if restart & gensincerestart>restartinterval if
trackovec(m,find(importance)) == trackovec(mrestartinterval,
find(importance)) x=randi(popsize,stringlength);
gensincerestart=0; end else gensincerestart=gensincerestart+1; end
end
%
***********************************************************
************
% HELPER FUNCTIONS
%**********************************************************
************** function [varargout]=getattributes(outputvec);
[a,b]=size(outputvec); for i = 1:nargout
varargout{i}=outputvec(:,i); end
%**********************************************************
************** function
[x,outputvec,fitness]=thresholdaccept(m,resettime,coolingrate,fitness,oldfitn
ess,x,xold,outputvec,outputvecold) popsize=length(fitness); for ii
= 1:popsize
if fitness(ii) > oldfitness(ii) %check to see if the child's fitness is worse
than the fitness of the same member of the old population
test=rand(1);
boltz=exp(-mod((m-1),resettime)*coolingrate); %if it is worse, accept
the lower performing individual according to the boltzmann
```

```
distribution
if test > boltz %reject the new (inferior) solution based on temperature
fitness(ii) = oldfitness(ii); %retain the previous (better) case
x(ii,:)=xold(ii,:); outputvec(ii,:)=outputvecold(ii,:); end end
end
%*******************************************************
************** function winnerindex=roulette(fitness,ncount)
popsize=length(fitness);
if isequal(ncount,[]) == 0 %use niching
fitness=fitness.*ncount'; end
rfitness=exp(-(fitness-min(fitness))); totalrfit=sum(rfitness);
rndn=totalrfit*rand(popsize,1); for kk = 1:popsize jjj=0; n=0;
while jjj < rndn(kk) n=n+1; jjj=jjj+rfitness(n); end
winnerindex(kk)=n; end
%*******************************************************
************** function winnerindex=tournament(fitness,ncount) if
isequal(ncount,[]) == 0
fitness=fitness.*ncount'; %use niching
end popsize=length(fitness);
list=[randperm(popsize)' randperm(popsize)']; %no replacement
tourn=fitness(list); if isequal(ncount,[]) == 0
ncount=ncount(list); end for j = 1:popsize
%populate the winner's
circle
if tourn(j,1) < tourn(j,2) %did number 1 win?
winnerindex(j)=list(j,1);
else if tourn(j,1) == tourn(j,2) & isequal(ncount,[]) == 0 %if theres a tie
and you're using niching, use niche count to break the tie if
ncount(j,1)<ncount(j,2) winnerindex(j)=list(j,1); else
winnerindex(j)=list(j,2); end else
winnerindex(j)=list(j,2); %2 won
end end end
%*******************************************************
************** function x1 = int2bin(int,bits,range)
[popsize,numvar]=size(int);
%if bits or range are singleton use repmat so that you assume all
%parameters are encoded equally
if max(size(bits)) == 1 bits=repmat(bits,numvar,1);
range=repmat(range,numvar,1); end for j = 1:popsize for k =
1:numvar int(j,k) =
(int(j,k)-range(k,1))*(2^bits(k)-1)/(range(k,2)-range(k,1)); end
end int=round(int); for j = 1:popsize for k = 1:numvar for i =
1:bits(k) pos = i+sum(bits(1:k-1)); if int(j,k)>=2^(bits(k)-i);
x1(j,pos)=1; int(j,k)=int(j,k)-2^(bits(k)-i); else x1(j,pos)=0;
end end end end
```

```
%*************************************************************
************** function int = bin2int(x,bits,range)
[popsize,stringlength]=size(x);
%if bits or range are singleton use repmat so that you assume all
%parameters are encoded equally
if max(size(bits)) == 1 numvar=stringlength/bits;
bits=repmat(bits,numvar,1); range=repmat(range,numvar,1); else
numvar=length(bits); end
%create 0 by 0 matrix
int(1:popsize,1:numvar)=repmat([range(1:numvar,1)]',popsize,1);
%calculate integers
for j = 1:popsize startposition=1; for k = 1:numvar cbits=bits(k);
pos=(startposition+cbits-1):-1:startposition;
startposition=startposition+cbits;
int(j,k)=int(j,k)+sum(2.^(find(x(j,pos))-1))*((range(k,2)-
range(k,1))/(2^cbits-1)); end end
%*************************************************************
************* function
[outputvec,targetinfo]=functioneval(functionname,executeindex,int,pswitch,
psettings,outputvec,model,mask)
if isequal(functionname,'stepwise') %use the passed in RS model for
function evaluation
betacoeffs=model.betacoeffs;,modelterms=model.modelterms;
[popsize,nvar]=size(int); executeindex=1:popsize; i=1:popsize;
[junk,nresponses]=size(betacoeffs);
secondorderterms=nchoosek([1:nvar]',2); globalinput=[int int.^2
int(:,secondorderterms(:,1)).*int(:,secondorderterms(:,2))]; for j
= 1:nresponses terms{j}=find(modelterms(:,j)); end
% evaluate RS model
for kk=1:popsize for j = 1:nresponses
[outputvec(kk,j)]=sum(betacoeffs{j}.*[1
globalinput(kk,terms{j})]'); end end
else if isequal(functionname,'kriging') %use the passed in Kriging model for
function evaluation outputvec=predictor(int,model.dmodel);
else if isequal(functionname,'interactive') %we are using the new
interactive genetic algorithm
[outputvec,targetinfo]=interactive(int,executeindex); else switch
pswitch
case 1 %PARALLEL EXECUTION
*********************** blocksize=pswitch;
executeindex=unique(executeindex); if exist('pfor') ~= 2 keyboard
end
pfor(1:max(size(executeindex)),['[ov(executeindex(%d),:)]='
functionname
'(int(executeindex(%d),:)'',executeindex(%d))'])
```

```
outputvec(executeindex,:)=ov(executeindex,mask);
%
pfor(1:blocksize:max(size(executeindex)),['[outputvec(executeindex(%d:mi
n(length(executeindex),%d+blocksize),:),compindex(execu
teindex(%d:min(length(executeindex),%d+blocksize)),:)]=' functionname
'(int(executeindex(%d:min(length(executeindex),%d+blocksize)),:)'',executei
ndex(%d:min(length(executeindex),%d+blocksize)))'])
case 2 %PARALLEL EXECUTION
*********************** executeindex=unique(executeindex);
aint=int(executeindex,:); [junk,numvar]=size(int); aindex=[1
max(size(executeindex)) 1 pswitch pswitch;2 1 1 0 numvar];
bindex=[1 max(size(executeindex)) 1 pswitch pswitch;2 1 1 0 1];
[poutputvec]=parallelize(psettings.ss,psettings.rs,inf,1,functionname,1,aint,a
index,executeindex,bindex); outputvec(executeindex,:)=poutputvec;
case 0 %NON PARALLEL EXECUTION
******************* executeindex=unique(executeindex);
i=1:max(size(executeindex));
[ov(executeindex(i),:)]=feval(functionname,int(executeindex(i),:),executeind
ex(i)); outputvec(executeindex,:)=ov(executeindex,mask); end end
end end
%
************************************************************
************ function
[paretox,paretoovec]=modelsearch(int,outputvec,numvar,bits,varrange,functi
onname,pswitch,psettings,Settings) linearrange=prctile(int,[5
95])';
includevars=find(linearrange(:,2)-linearrange(:,1)); %these variables with be
used for regression
excludevars=setdiff(1:numvar,includevars); %these ones wont, their values
will be fixed at the mean value
regressionint=int(find(~isnan(outputvec(:,1))),:); %use only parameters that
have significant variation for regression
regressionoutputvec=outputvec(find(~isnan(outputvec(:,1))),:); %this is used
to get rid of failed cases [junk,numinputs]=size(regressionint);
[junk,numoutputs]=size(regressionoutputvec); switch
Settings.surrogatetype case 'stepwise'
for j = 1:numoutputs %find betacoeffs coefficients
interaction(:,j)=[1:numvar]';
terms=nchoosek(interaction(:,j),2); %we will include 2nd order
interactions
badinteractions=find(any(ismember(terms',excludevars))); %but not
the ones that are excluded due to lack of variability (they would
cause singularities
inter=regressionint(:,terms(:,1)).*regressionint(:,terms(:,2));
stepint=[regressionint regressionint.^2 inter]; %linear, quadratic, and
```

```
2nd order terms included excluded=zeros(1,2*numvar+length(terms));
excluded([excludevars excludevars+numvar
2*numvar+badinteractions])=1;
[junk,SE,PVAL,IN,STATS,NEXTSTEP,HISTORY]=stepwisefit(stepint,reg
ressionoutputvec(:,j),'penter',.03,'premove',.05,'keep',exclude
d,'maxiter',400); %which terms?
INMODEL(:,j)=IN';
betacoeffs{j}=regress(regressionoutputvec(:,j),[ones(length(regressionint),1)
stepint(:,find(INMODEL(:,j)))]);
%fit linear model
end case 'kriging'
[regressionint,ix]=unique(regressionint,'rows');
regressionoutputvec=regressionoutputvec(ix,:);
[regressionint,regressionoutputvec]=(dsmerge(regressionint,regressionoutput
vec,.5));
dmodel=dacefit(regressionint,regressionoutputvec,@regpoly2r,@corrgauss,
10,.000001,20);%10*ones(1,numinputs),.0001*ones(1,numinp
uts),100*ones(1,numinputs)); end
% set ga parameters
Settings.fittype='MO'; %SO for single objective, MO for multi objective
Settings.rangeshift=0;
Settings.importance=Settings.importance(find(Settings.importance));
Settings.modelsearch=0; %else infinite loop
Settings.localsearch=0; Settings.niching=0; Settings.savetomat=0;
Settings.surrogatesearch=0;
Settings.functionname=Settings.surrogatetype;
Settings.varrange=varrange(includevars,:);
Settings.munvar=length(varrange); Settings.numgen=100;
Settings.popsize=Settings.popsize/2; Settings.elitism=1;
replacement.switch = 'BF'; switch Settings.surrogatetype case
'stepwise' Settings.model.betacoeffs=betacoeffs;
Settings.model.modelterms=INMODEL; case 'kriging'
Settings.model.dmodel=dmodel; end
%run GA
[x,outputvec,fitness,paretoint]=runga(Settings); %receive the suggested
insertion points back from the GA
paretoint(:,excludevars)=mean(int(:,excludevars));
paretox=encode(paretoint,Settings.encoding,Settings.numconfigvars,bits,var
range); [lengthpi,widthpi]=size(paretoint);
[paretoovec]=functioneval(functionname,[1:lengthpi]',paretoint,pswitch,pset
tings,[],[]); %evaluate the fitness of the suggested
designs with the full design
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
```

```
function out = randi(cols,rows,interval); %function to produce random
integers if nargin == 2 out=floor(2*rand(cols,rows)); else
out=repmat(interval(1),cols,rows);%interval(1)*ones(cols,rows);
out=out+floor((interval(2)-interval(1)+1)*rand(cols,rows)); end
%
************************************************************
************ function
[X1out,X2out,indices]=crossvec(X1,X2,crosstype)
[junk,strlength]=size(X1); %find out string length
switch crosstype case '2pt' X1out=X1; X2out=X2;
start = randi(1,1,[1,strlength]); %finds start position
wrap = randi(1,1,[1,strlength - 1]); %finds interval
indices=mod(start:(start+wrap),strlength);
X1out(indices)=X2(indices); X2out(indices)=X1(indices); case
'uniform' X1out=X1; X2out=X2; indices=randi(1,strlength);
indices=find(indices); X1out(indices)=X2(indices);
X2out(indices)=X1(indices); case '2ptlinear' X1out=X1; X2out=X2;
start = randi(1,1,[1,strlength]); %finds start position
wrap = randi(1,1,[1,strlength - 1]); %finds interval
indices=mod(start:(start+wrap),strlength);
range=X2(indices)-X1(indices);
alpha=-.25+1.5*rand(1,length(indices));
X1out(indices)=X1out(indices)+alpha.*range;
alpha=-.25+1.5*rand(1,length(indices));
X2out(indices)=X1out(indices)+alpha.*range; case 'uniformlinear'
X1out=X1; X2out=X2; indices=randi(1,strlength);
indices=find(indices); if ~isequal(indices,[])
range=X2(indices)-X1(indices);
alpha=-.25+1.5*rand(1,length(indices));
X1out(indices)=X1out(indices)+alpha.*range;
alpha=-.25+1.5*rand(1,length(indices));
X2out(indices)=X1out(indices)+alpha.*range; end end
%
************************************************************
************ function
[paretox,paretoovec,paretofit]=hillclimb(xin,outputvecin,fitin,bits,varrange,f
unctionname,pswitch,psettings,numsearches,outputv
ect,importance,fittype,MOtype,encoding,numconfigvars);
%hillclimbing
disp('hillclimbing') [popsize,stringlength]=size(xin);
paretox=xin; paretoovec=outputvecin; paretofit=fitin;
for kk = 1:numsearches %number of hillclimbs to perform
for jj = 1:popsize
u = randi(1,1,[numconfigvars,stringlength]); %random
number to find mutation location switch encoding case 'binary'
```

```
xin(jj,u)=~xin(jj,u); case 'real'
xin(jj,u)=xin(jj,u)+normrnd(0,.1); end end
tryint=decode(xin,encoding,numconfigvars,bits,varrange);
[outputvectest]=functioneval(functionname,1:popsize,tryint,pswitch,[],[]);
%fitness transformation (use weighted sum, pareto fitness, etcera)
switch fittype case 'MO'
newfitness=MOfitness(outputvectest,outputvect,importance,MOtype);
case 'SO' newfitness=outputvectest(:,length(outputvectest(1,:)));
end for ll = 1:popsize if newfitness(ll) > paretofit(ll)
disp('Change rejected') else disp('Change accepted')
paretox(ll,:)=xin(ll,:); paretoovec(ll,:)=outputvectest(ll,:);
paretofit(ll)=newfitness(ll); end fittrack(kk)=paretofit; end
figure(2) plot(fittrack) drawnow xin=paretox; end
%*******************************************************
************** function ncount=nichecount(outputvec,nicheradius)
%normalize everyone
maximum=max(outputvec); minimum=min(outputvec);
psize=length(outputvec); [junk,pwidth]=size(outputvec);
normalized=(outputvec-repmat(minimum,psize,1))./(repmat(maximumminimum,
psize,1)); dist=ones(psize); for i = 1:psize for j = 1:psize
dist(i,j)=(sum((normalized(i,:)-normalized(j,:)).^2)).^.5; end end
ncount=sum(dist<nicheradius,2)';
%
*******************************************************
******** function
int=decode(x,encoding,numconfigvars,bits,varrange)
[popsize,stringlength]=size(x);
numvar=numconfigvars+length(varrange);
continuousx=x(:,numconfigvars+1:stringlength); switch encoding
case 'binary'
int(:,(numconfigvars+1):numvar)=bin2int(continuousx,bits,varrange);
%convert binary string to
decimal case 'real'
int(:,(numconfigvars+1):numvar)=repmat(varrange(:,1)',popsize,1)+continuo
usx.*repmat(varrange(:,2)'-varrange(:,1)',popsize,1); end if
numconfigvars>0 int(:,1:numconfigvars)=x(:,1:numconfigvars); end
%
*******************************************************
********* function x =
encode(int,encoding,numconfigvars,bits,varrange)
[popsize,numvar]=size(int);
continuousint=int(:,numconfigvars+1:numvar); switch encoding case
'binary'
x(:,(numconfigvars+1):numvar)=int2bin(continuousint,bits,varrange);
%convert continuous to
```

```
binary case 'real'
x(:,(numconfigvars+1):numvar)=(continuousintrepmat(
varrange(:,1)',popsize,1))./repmat(varrange(:,2)'-
varrange(:,1)',popsize,1) ; end if numconfigvars>0
x(:,1:numconfigvars)=int(:,1:numconfigvars); end
%
*******************************************************
********* function
[paretox,paretoovec,paretofit]=sqp(xin,outputvecin,fitin,bits,varrange,functi
onname,pswitch,psettings,numsearches,outputvect,im
portance,fittype,MOtype,encoding,numconfigvars);
%hillclimbing
disp('hillclimbing') [popsize,stringlength]=size(xin);
paretox=xin; paretoovec=outputvecin; paretofit=fitin;
%func = @(x)
MOfitness(functioneval(functionname,1:popsize,[xin(1:numconfigvars)
x],pswitch,[],[]),outputvect,importance,MOtype); func = @(x)
functioneval(functionname,1:popsize,[xin(1:numconfigvars)
x],pswitch,[],[])
options=optimset('DiffMaxChange',.1,'DiffMaxChange',.01,'MaxFunEvals',3
00)
fgoalattain(func,paretox((numconfigvars+1):end),outputvect,importance,[],[]
,[],[],zeros(length(paretox)),ones(length(paretox)), [],options)
for kk = 1:numsearches %number of hillclimbs to perform
for jj = 1:popsize
u = randi(1,1,[numconfigvars,stringlength]); %random
number to find mutation location switch encoding case 'binary'
xin(jj,u)=~xin(jj,u); case 'real'
xin(jj,u)=xin(jj,u)+normrnd(0,.1); end end
tryint=decode(xin,encoding,numconfigvars,bits,varrange);
[outputvectest]=functioneval(functionname,1:popsize,tryint,pswitch,[],[]);
%fitness transformation (use weighted sum, pareto fitness, etcera)
switch fittype case 'MO'
newfitness=MOfitness(outputvectest,outputvect,importance,MOtype);
case 'SO' newfitness=outputvectest(:,length(outputvectest(1,:)));
end for ll = 1:popsize if newfitness(ll) > paretofit(ll)
disp('Change rejected') else disp('Change accepted')
paretox(ll,:)=xin(ll,:); paretoovec(ll,:)=outputvectest(ll,:);
paretofit(ll)=newfitness(ll); end fittrack(kk)=paretofit; end
figure(2) plot(fittrack) drawnow xin=paretox; end
% ****************************************************
function dens=crowded(categorytypes,intt,x,outputvec)
numconfigvars=length(categorytypes);
%calculate the percentage of the population of each individual
for i = 1:numconfigvars for j = 1:length(categorytypes{i})
```

```
per{i}(j)=sum(intt(:,i)==categorytypes{i}(j))/size(intt,1); end
end
%find the distance (in response space) between individuals
maximum=max(outputvec); minimum=min(outputvec);
outputvec(isnan(outputvec(:,1)),:)=repmat(maximum,sum(isnan(outputvec(:,
1))),1); normalized=(outputvecrepmat(
minimum,size(outputvec,1),1))./(repmat(maximumminimum,
size(outputvec,1),1)); dist=distmat1(normalized);
dist(logical(eye(size(x,1))))=inf; dist=dist+1e-2;
distindex=round(size(outputvec,1)^.5); [a,b]=sort(dist); for i =
1:size(x,1)
%iseq(i,1)=sum(all(repmat(x(i,1:length(categorytypes)),size(x,1),1)==x(:,1:l
ength(categorytypes)),2)); for j = 1:numconfigvars
scarse(i,j)=length(categorytypes(j))*per{j}(categorytypes{j}==x(i,j));
end
iseq(i,1)=sum(sum(repmat(x(i,1:length(categorytypes)),size(x,1),1)==x(:,1:l
ength(categorytypes)),2).*1./dist(:,i));
%iseq(i,1)=sum(sum(repmat(x(i,1:length(categorytypes)),distindex,1)==x(b(
1:distindex,i),1:length(categorytypes))).*scarse(i,:),
2);%.*1./dist(b(1:distindex,i),i));
end dens=1/numconfigvars*sum(scarse,2)+iseq;
%
*************************************************************
********* function
[xout,ovout,paretofit]=powellsearch(x,outputvec,fitness,bits,varrange,functi
onname,pswitch,psettings,numsearches,w,encoding,num
configvars,mask)
%hillclimbing
disp('powell searching') [popsize,stringlength]=size(x);
% switch pswitch
% case 1 %PARALLEL EXECUTION
************************
% blocksize=pswitch;
% executeindex=1:popsize;
% pfor(executeindex',['[xout(%d,:)]=powell(''' functionname
''',x(%d,:),1:numconfigvars,mask,(%d),w(%d,:));'])
%
ovout=functioneval(functionname,1:popsize,xout,pswitch,psettings,[],[],mas
k);
% case 0 %NON PARALLEL EXECUTION
******************* for i = 1:popsize
xout(i,:)=powell(functionname,x(i,:),1:numconfigvars,mask,i,w(i,:));
end
ovout=functioneval(functionname,1:popsize,xout,0,psettings,[],[],mask);
% end
```

```
paretofit=[];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
function [x,y]=swapvals(x,y,ind) temp=x; x(ind)=y(ind);
y(ind)=temp(ind);
```

APPENDIX C

ADDITIONAL VALIDATION RESULTS

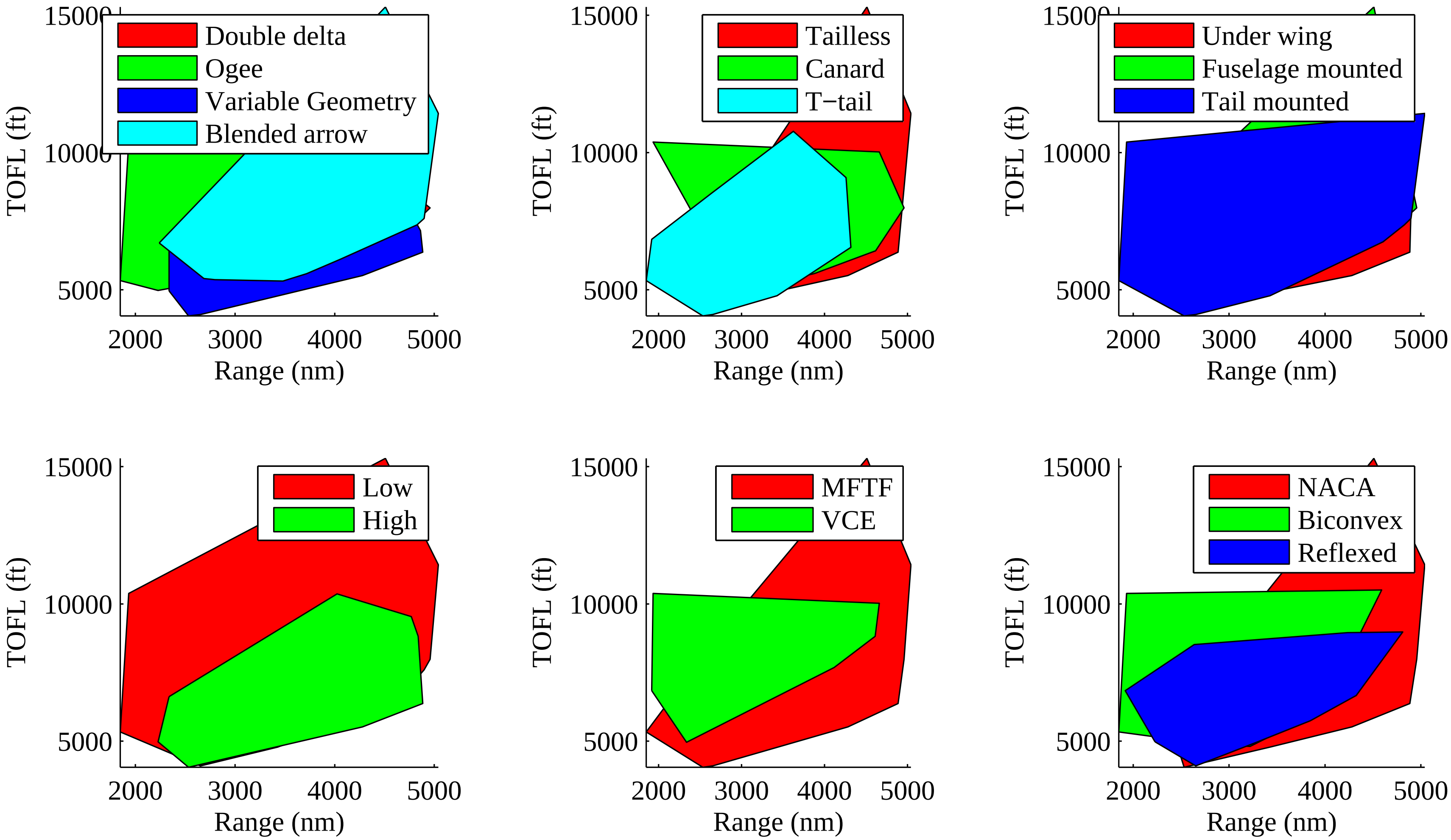


Figure 104: Impact of configuration alternatives on range and takeoff field length

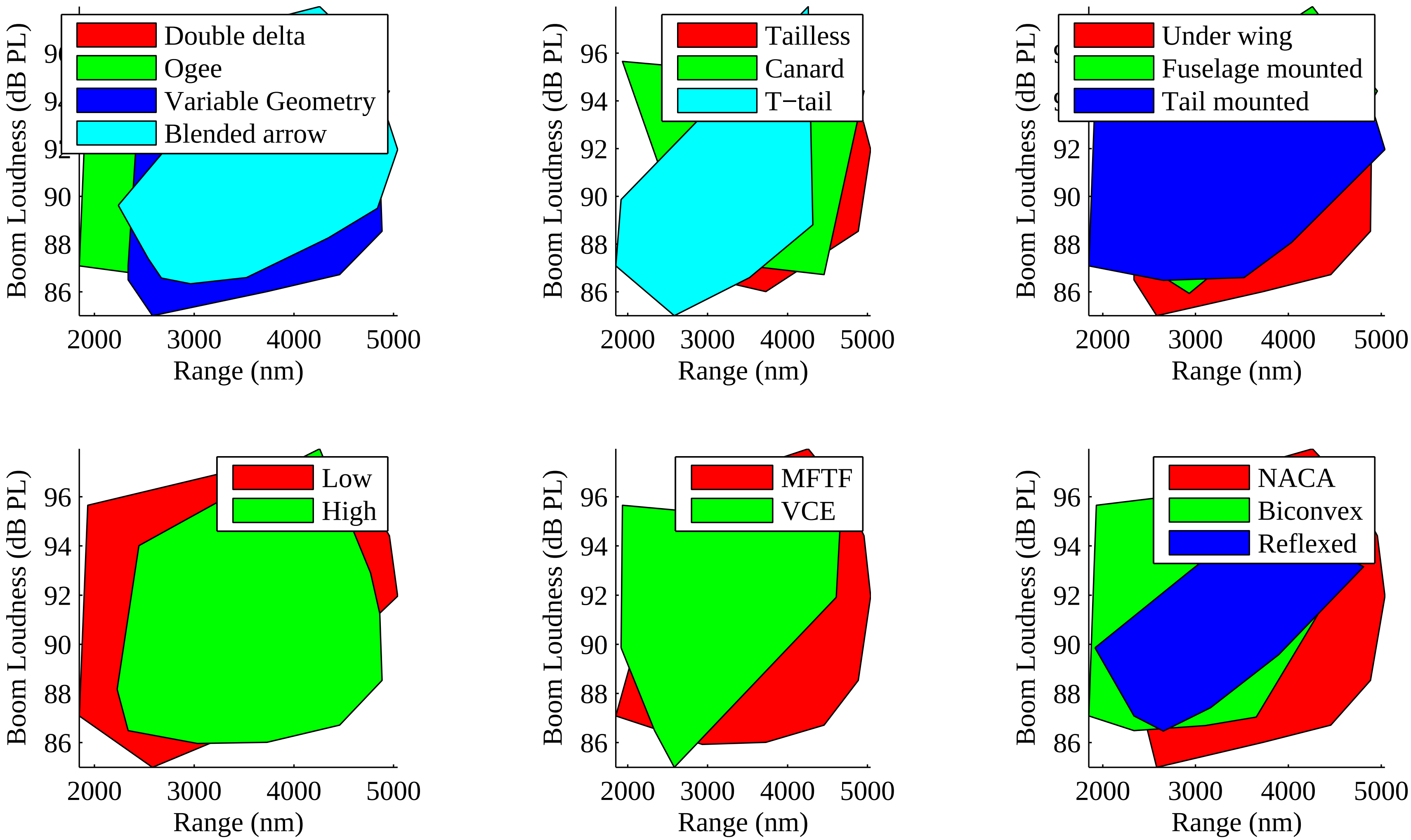


Figure 105: Impact of configuration alternatives on range and sonic boom loudness

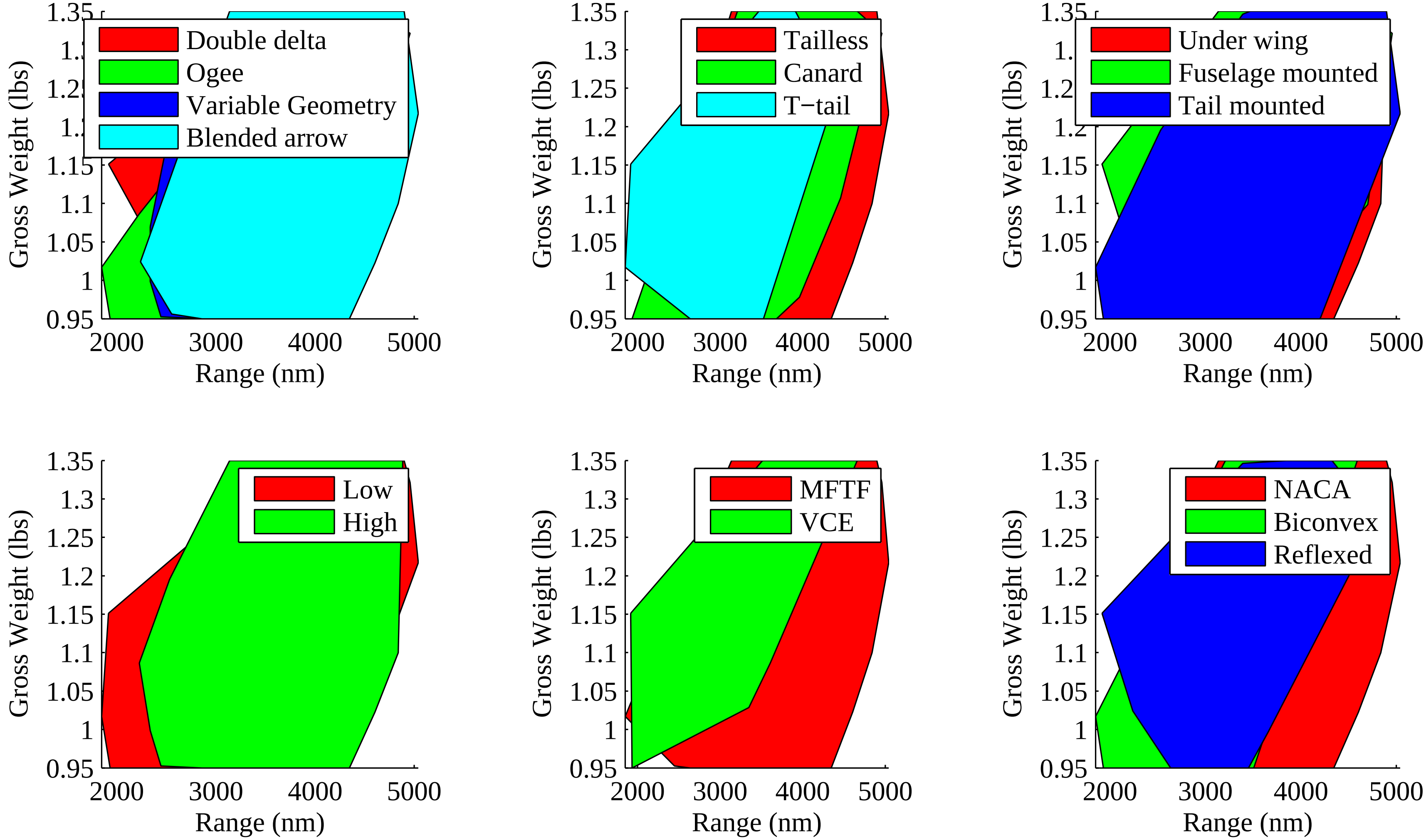


Figure 106: Impact of configuration alternatives on range and takeoff gross weight

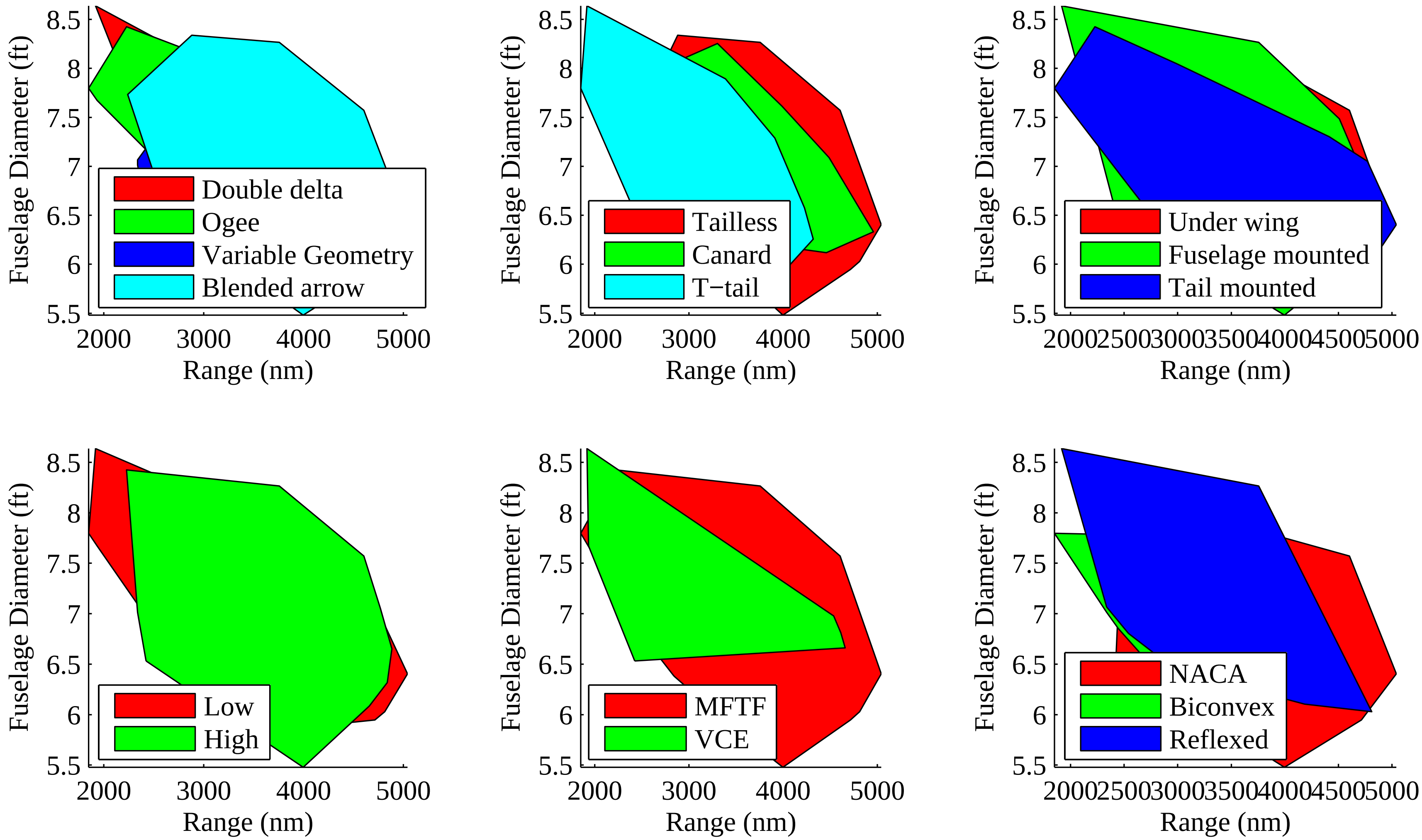


Figure 107: Impact of configuration alternatives on range and fuselage diameter

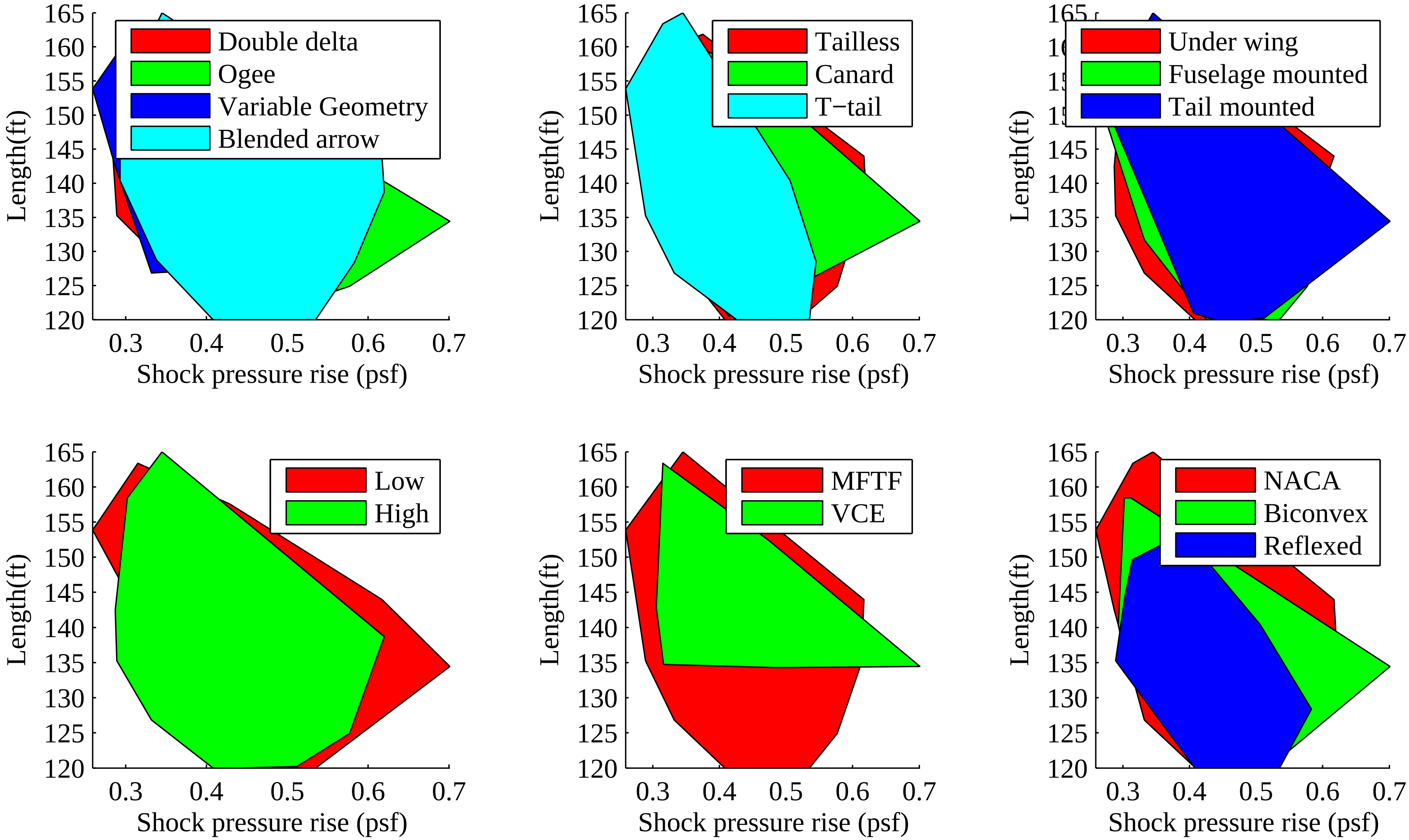


Figure 108: Impact of configuration alternatives on shock pressure rise and length

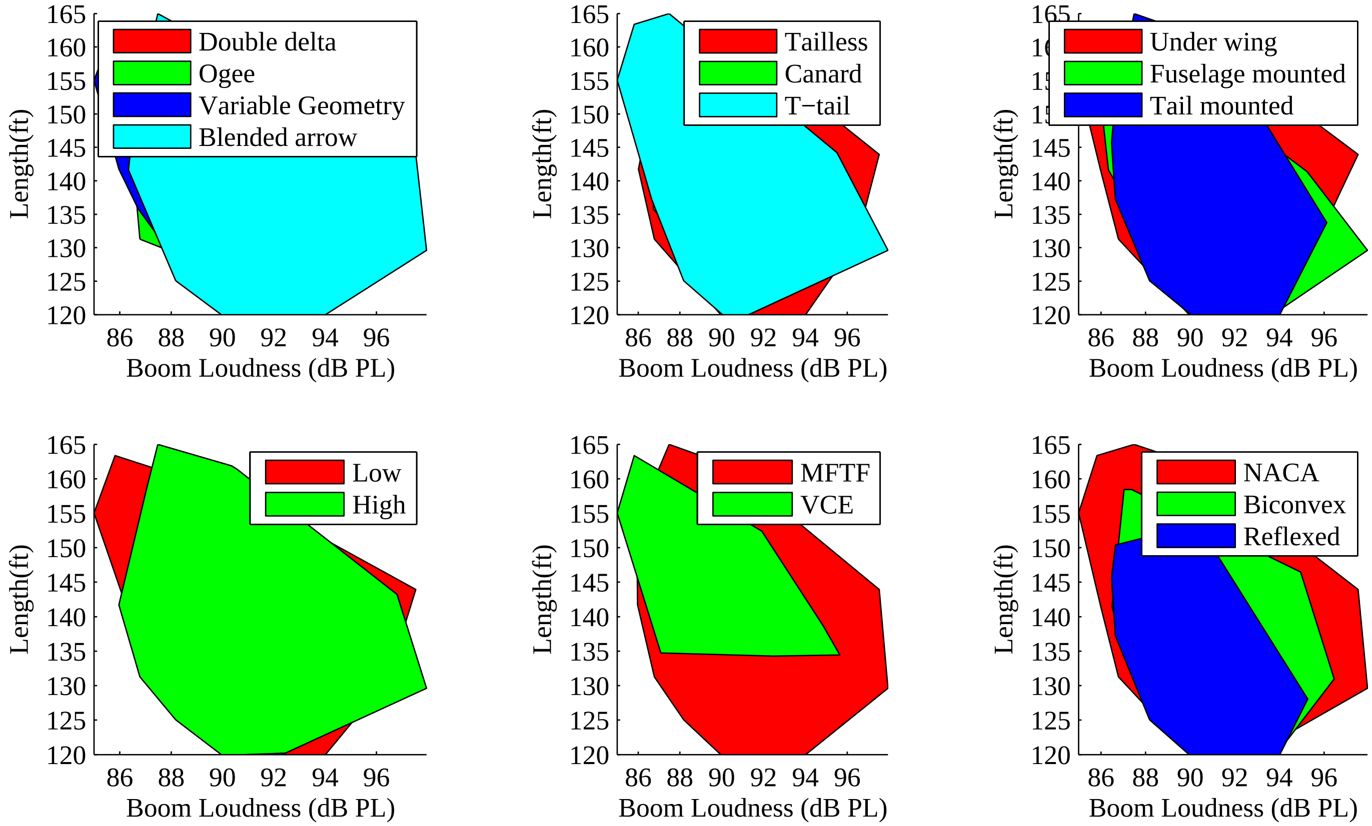


Figure 109: Impact of configuration alternatives on sonic boom loudness and length

REFERENCES

[1] Aeronautics and Space Engineering Board, *Commercial Supersonic Technology: The Way Ahead*. Washington, D.C.: National Academies Press, 2001.

[2] Altschuller, G., *Creativity as an Exact Science*. New York: Gordon and Breach, 1988.

[3] Anderson, J. Y. R. and Yurkovich, R., "A Description of the F/A-18E/F Design and Design Process," 1998. AIAA Paper 98-4701.

[4] Anonymous, "Supersonic flow," 2000. http://www.centennialofflight.gov.

[5] Aronstein, D., "Clean-sheet design and initial optimization in non-traditional multidisciplinary design problems," Tech. Rep. AIAA-2004-4602, 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Jan. 2005.

[6] Aronstein, D. and Schueler, K., "Conceptual design of a sonic boom constrained supersonic business aircraft," Tech. Rep. AIAA-2004-697, 42nd AIAA Aerospace Sciences Meeting and Exhibit, Jan. 2004.

[7] Back, T., "Optimal mutation rates in genetic search," in *Proceedings of the 5th International Conference on Genetic Algorithms*, (San Francisco, CA, USA), pp. 2–8, Morgan Kaufmann Publishers Inc., 1993.

[8] Baker, A., *The Role of Mission Requirements, Vehicle Attributes, Technologies and Uncertainty in Rotorcraft System Design*. PhD thesis, Georgia Institute of Technology, 2002.

[9] Beissner, F., Lovell, W., Robins, A., and Swanson, E., "Application of Near-Term Technology to a Mach 2.0 Variable-Sweep Wing, Supersonic-Cruise Executive jet," Tech. Rep. CR-172321, National Air and Space Administration, 1984.

[10] Bentley, P., *Evolutionary Design by Computers*, ch. From coffee tables to Hospitals: Generic Evolutionary Approach. San Francisco, CA: Morgan Kaufmann, 1999.

[11] Bentley, P., *Evolutionary Design by Computers*. San Francisco, CA: Morgan Kaufmann, 1999.

[12] Biles, J., "Genjam: A genetic algorithm for generating jazz solos," in *ICMC'94*, (Aarhus, Denmark), 1994.

[13] Biles, J., Anderson, P., and Loggi, W., "Neural network fitness functions for a musical IGA," *IIA'96/SOCO'96*, 1996.

[14] Boggs, P. T. and Tolle, J. W., "Sequential quadratic programming," in *Acta Numerica*, pp. 1–51, 1995.

[15] Boppe, C. W., *Engineering Methods in Aerodynamic Analysis and Design of Aircraft.* AGARD, 1992.

[16] Borer, N. and Mavris, D., "Formulation of a multi-mission sizing methodology for competing configurations," in *42nd Aerospace Sciences Meeting and Exhibit*, (Reno, NV), AIAA, 2004.

[17] Bos, D., "Multidisciplinary design optimization of a supersonic transport configuration using a hybrid genetic/gradient-based method," in *6th AIAA/-NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, (Washington, D.C.), AIAA, 1996.

[18] Branke, J., Schmeck, H., Deb, K., and Reddy, S., "Parallelizing multi-objective evolutionary algorithms: Cone separation," Tech. Rep. KanGAL Report No. 2004017, Kanpur, India, 2004.

[19] Braun, R., *Collaborative Optimization: An architecture for large-scale distributed design.* PhD thesis, Stanford University, 1996.

[20] Briceno, S., Buonanno, M., Fernandez, I., and Mavris, D., "A parametric exploration of supersonic business jet concepts utilizing response surfaces," tech. rep., AIAA Aviation, Technology, Integration, and Operations (ATIO) Conference, 2002.

[21] Bruner, H. S., "SSBJ: A Technical Challenge," *ICAO Journal*, Aug. 1991.

[22] Byrne, D. and Taguchi., S., "The Taguchi approach to parameter design," in *40th Annual Quality Congress Transactions*, 1987.

[23] Caldwell, C. and Johnston, V., "Tracking a criminal suspect through "face-space" with a genetic algorithm," in *Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann, 1991.

[24] Chen, J.-H., "Theoretical analysis of multi-objective geneticalgorithms - convergence time, population sizing,and disequilibrium," 2003. REPORT FOR IEEE NNS WALTER KARPLUS RESEARCH GRANT.

[25] Chung, H. and Alonso, J., "Multiobjective optimization using approximation model-based genetic algorithms," in *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, (Albany, NY), 2004.

[26] Coen, P., "Development of a computer technique for the prediction of transport aircraft flight profile sonic boom signatures," Master's thesis, The George Washington University, 1991.

[27] Coen, P., "Vehicle systems program supersonic aircraft sector overview," 2004. Presentation given at 1st VSP Conference, Atlanta, Ga.

[28] Committee, A. M. T., "Current state of the art on multidisciplinary design optimization (mdo)," Tech. Rep. ISBN 1-56347-021-7, American Institute of Aeronautics and Astronautics, Sept. 1991.

[29] Craidon, C., "Description of a digital computer program for airplane configuration plots," Tech. Rep. TM X-2074, National Air and Space Administration, 1970.

[30] Cvetkovic, D. and Parmee, I., "Preferences and their application in evolutionary multiobjective optimisation," *IEEE Transactions on Evolutionary Computation*, vol. 6(1), pp. 42–57, 2002.

[31] Darden, C., "Sonic boom minimization with nose-bluntness relaxation," Tech. Rep. NASA TP-1348, Hampton, VA, 1979.

[32] Das, I. and Dennis, J., "A Closer Look at Drawbacks of Minimizing Weighted Sums of Objectives for Pareto Set Generation in Multicriteria Optimization Problems," *Structural Optimization*, vol. 14, no. 1, 1997.

[33] Dasgupta, D. and McGregor, D., "Sga: A structured genetic algorithm," 1992.

[34] Dawkins, R., *The Blind Watchmaker*. Essex, U.K.: Longman, 1986.

[35] De, A., "Pugh matrix," 2004. http://www.isixsigma.com/dictionary/PughMatrix-384.htm.

[36] De Jong, K., *An analysis of the behavior of a class of genetic adaptive systems.* PhD thesis, University of Michigan, 1975.

[37] Deb, K., "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 2, 1995.

[38] Deb, K., *Multi-Objective Optimization using Evolutionary Algorithms*. West Sussex, U.K.: Wiley and Sons. inc., 2002.

[39] Deb, K., Agrawal, S., Pratab, A., and Meyarivan, T., "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II," in *Proceedings of the Parallel Problem Solving from Nature VI Conference* (Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J. J., and Schwefel, H.-P., eds.), (Paris, France), pp. 849–858, Springer. Lecture Notes in Computer Science No. 1917, 2000.

[40] Deb, K. and Beyer, H.-G., "Self-adaptation in real-parameter genetic algorithms with simulated binary crossover," in *Proceedings of the Genetic and Evolutionary Computation Conference* (Banzhaf, W., Daida, J., Eiben, A. E.,

Garzon, M. H., Honavar, V., Jakiela, M., and Smith, R. E., eds.), vol. 1, (Orlando, Florida, USA), pp. 172–179, Morgan Kaufmann, 13-17 1999.

[41] Demis, R., "Gulfstream and Sukhoi go Supersonic," *Aerospace America*, vol. 28, 1990.

[42] Department of Defense, *Systems Engineering Fundamentals*. Fort Belvoir, Virginia: Defense Acquisition University Press, 2001.

[43] Dieter, G., *Engineering Design*. New York, NY: McGraw Hill Inc., 2000.

[44] Dutton, G., "Preliminary studies of a supersonic business jet," Tech. Rep. SAE 670276, Business Aircraft Conference, 1967.

[45] Eby, D., "An injection island ga for flywheel design optimization," in *Proceedings of the 5th European Congress on Intelligent Techniques and Soft Computing*, 1997.

[46] Engineous Software, "iSight product information webpage," 2005. http://www.engineous.com/product

[47] Esbensen, H., "Explorer: An interactive floorplanner for design space exploration," in *European Design Automation Conference*, (Geneva, Switzerland), 1996.

[48] Eshelman, L. J. and Schaffer, J. D., "Real coded genetic algorithmms and interval schemata," in *Foundations of Genetic Algorithms 2*, 1993.

[49] Eyer, J., "Modifications to the noise prediction capabilities of pboom," 2004. Aerospace Systems Design Lab.

[50] Fang, R., "2d free hand recognition system," Master's thesis, Oregon State University, 1988.

[51] Federal Aviation Administration, "Supersonic aircraft noise," 2004. http://www.faa.gov/about/office5Fnoise/.

[52] Fenbert, J., Ozoroski, L., Geiselhart, K., Shields, W., and McElroy, M., "Concept Development of a Mach 2.4 High-Speed Civil Transport," Tech. Rep. NASA/TP-1999-209694, Hampton, VA, 1999.

[53] Fletcher, R., *Practical Methods of Optimization*. John Wiley & Sons, 2000.

[54] Fletcher, R. and Reeves, C., "Function minimization by conjugate gradients," *The Computer Journal*, vol. 7, no. 3, pp. 149–154, 1964.

[55] Flug Revue, "Dassault supersonic business jet," 2005. http://www.flug-revue.rotor.com/FRTypen/FRDasSSB.htm.

[56] Fonseca, C. M. and Fleming, P. J., "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization," in *Genetic Algorithms: Proceedings of the Fifth International Conference*, pp. 416–423, Morgan Kaufmann, 1993.

[57] Geoffrion, A. M., Dyer, J. S., and Feinberg, A., "An interactive approach for multi-criterion optimization with an application to the operation of an academic department," *Management Science*, vol. 19, no. 4, 1972.

[58] Giesing, J. and Barthelemy, J., "Summary of Industry MDO Applications and Needs," Tech. Rep. AIAA-98-4737, 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Sept 1998.

[59] Glover, F., "Future paths for integer programming and links to artificial intelligence," *The Computer Journal*, vol. 5, pp. 533–549, 1986.

[60] Goldberg, D., *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley, 1989.

[61] H. Carlson J. Chu L. Ozoroski and A. McCullers, "Guide to AERO2S and WINGDES Computer Codes for Prediction and Minimization of Drag Due to Lift," Tech. Rep. TP 3637, National Air and Space Administration, Nov. 1980.

[62] Harik, G. R., "Finding multimodal solutions using restricted tournament selection," in *Proceedings of the Sixth International Conference on Genetic Algorithms* (Eshelman, L., ed.), (San Francisco, CA), pp. 24–31, Morgan Kaufmann, 1995.

[63] Harris, R., "An analysis correlation of aircraft wave drag," Tech. Rep. TMX-947, National Air and Space Administration, Mar. 1964.

[64] Hayes, W. D., "Linearized supersonic flow," Tech. Rep. AL-222, North American Aviation, 1947.

[65] Henne, P., "Case for small supersonic civil aircraft," *Journal of Aircraft*, vol. 42, no. 3, 2005.

[66] Heppenheimer, T., *The Space Shuttle Decision*. Washington, D.C.: NASA Press, 1999.

[67] Ho, Y.-C., "An explanation of ordinal optimization: Soft computing for hard problems," *Information Sciences*, vol. 113, 1999.

[68] Holland, J., *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Cambridge, MA: MIT Press, 1992.

[69] Hollingsworth, P., *Requirements Controlled Design: A Method for Discovery of Discontinuous System Boundaries in the Requirements Hyperspace*. PhD thesis, Georgia Institute of Technology, 2004.

[70] Horn, J., Nafpliotis, N., and Goldberg, D. E., "A Niched Pareto Genetic Algorithm for Multiobjective Optimization," in *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, vol. 1, (Piscataway, New Jersey), pp. 82–87, IEEE Service Center, 1994.

[71] Howe, D., "Engine placement for sonic boom mitigation," Tech. Rep. AIAA-2002-148, 40th AIAA Aerospace Sciences Meeting and Exhibit, Jan. 2002.

[72] Howe, D., "Improved sonic boom minimization with extendable nose spike," Tech. Rep. AIAA-2005-1014, 43rd AIAA Aerospace Sciences Meeting and Exhibit, Jan. 2005.

[73] Huang, X., Haftka, R., Grossman, B., and Mason, W., "Comparison of statistical-based weight equations with structural optimization for supersonic transport wings," Tech. Rep. AIAA 94-4379, 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 1994.

[74] Hwang, C. and Yoon, K., *Multiple Attribute Decision Making: Methods and Applications:A State-of-the-ArtSurvey*. Springer-Verlag, 1981.

[75] Ignizio, J., *Goal Programming and Extensions*. Lexington Books, 1976.

[76] Inoue, T., Furuhashi, T., and Fujii, M., "Development of a nurse scheduling support system using interactive EA," *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 5, 1999.

[77] Ishibuchi, H. and Shibata, Y., "An empirical study on the effect of mating restriction on the search ability of emo algorithms," in *EMO*, pp. 433–447, 2003.

[78] Jenkins, D. and Matrin, R., "The importance of free hand sketching in conceptual design: Automatic sketch input," in *ASME Conference on Design theory and Methodology (DTM93)*, 1993.

[79] Johnson Space Center, "CLIPS '94," in *Third Conference on CLIPS Proceedings*, 1994.

[80] Jones, R. T., "Theory of wing-body drag at supersonic speeds," Tech. Rep. 1284, NACA, Jan. 1956.

[81] Karnozov, V., "Boeing and sukhoi consider supersonic alliance," 1999. http://www.aeroworldnet.com/1ra10189.htm.

[82] Khare, V., "Performance scaling of multi-objective evolutionary algorithms."

[83] Kidwell, G. and Eskey, M., "Expert systems and their use in augmenting design optimization," Tech. Rep. AIAA-85-3095, Aircraft Design Systems and Operations Meeting, Oct. 1985.

[84] Kim, H. and Cho, S., "Application of interactive genetic algorithm to fashion design," *Engineering Applications of Artificial Intelligence*, vol. 13, no. 6, 2000.

[85] Kirby, M., *A Methodology for Technology Identification, Evaluation, and Selection in Conceptual and Preliminary Aircraft Design*. PhD thesis, Georgia Institute of Technology, 2001.

[86] Kirschbaum, N., "Aircraft layout," 2002. http://www.aoe.vt.edu/ mason/-Mason

[87] Komadina, S., Drake, A., and Bruner, S., "Development of a quiet supersonic aircraft with technology applications to military and civil aircraft," Tech. Rep. AIAA-2002-519, American Instiitute of Aeronautics and Astronautics, Jan. 2002.

[88] Kroo, I., *Aircraft Design: Synthesis and Analysis*. Stanford, CA: Desktop Aeronautics, 2004.

[89] Kroo, I., "An interactive system for aircraft design and optimization," Tech. Rep. AIAA-92-1190, Aerospace Design Conference, Feb. 1992.

[90] Leatherwood, J. and Sullivan, B., "Laboratory study of effects of sonic boom shaping on subjective loudness and acceptability," Tech. Rep. TP-3269, NASA, 1992.

[91] Leger, C., *Automatic Synthesis and Optimization of Robot Configurations: An Evolutionary Approach*. PhD thesis, Carnegie Mellon University, 1999.

[92] Liljedahl, U., *Software development supported by QFD*. PhD thesis, Lulea Tekniska Universitet, 2001.

[93] Luenberger, D., *Linear and Nonlinear Programming*. Addison-Wesley, 1984.

[94] Lyman, V. and Morgenstern, J., "Calculated and measured pressure fields for an aircraft designed for sonic-boom alleviation," Tech. Rep. AIAA-2004-4846, American Instiitute of Aeronautics and Astronautics, Aug. 2004.

[95] Mack, R., "An analysis of measured sonic-boom pressure signatures from a langley wind-tunnel model of a supersonic-cruise business jet concept," Tech. Rep. NASA TM-2003-212447, Hampton, VA, 2003.

[96] Maglieri, D. J., "A compilation and review of supersonic business jet studies," Tech. Rep. 96.1, Eagle Aeronautics, Mar. 1996.

[97] Malinchik, O. B. S., “A broad and narrow approach to interactive evolutionary design an aircraft design example,” *Lecture Notes in Computer Science*, vol. 3103, 2004.

[98] Marx, W., Mavris, D., and Schrage, D., “Effects of alternative wing structural concepts on high speed civil transport life cycle costs,” in *37th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, (Salt Lake City, UT), AIAA, 1996.

[99] Mascitti, V. R., “A preliminary study of the performance and characteristics of a supersonic executive aircraft,” Tech. Rep. TM-74055, National Air and Space Administration, Sept. 1977.

[100] Mattson, C. and Messac, A., “Concept selection using s-pareto frontiers,” *AIAA Journal*, vol. 41, no. 6, 2003.

[101] Mavris, D., Baker, A., and Schrage, D., “IPPD Through Robust Design Simulation for an Affordable Short Haul Civil Tiltrotor,” in *American Helicopter Society 53rd Annual Forum*, 1997.

[102] Mavris, D., “Advanced design methods class notes,” 2001. AE 6341, Georgia Institute of Technology.

[103] Mavris, D. and DeLaurentis, D., “Methodology for examining the simultaneous impact of requirements, vehicle characteristics, and technologies on military aircraft design,” in *22nd Congress of the International Councilon the Aeronautical Sciences (ICAS)*, (Harrogate, England), 2000.

[104] McCullers, A., “Flight optimization system,” 2001. NASA Langley Research Center, Hampton, VA.

[105] McMasters, J., “Some thoughts on design education,” Tech. Rep. AIAA-85-3079, American Instiitute of Aeronautics and Astronautics, Oct. 1985.

[106] Meisel, W., *Multiple Criteria Decision Making*, ch. Tradeoff decision in multiple criteria decision-making. Columbia, SC: USC Press, 1973.

[107] Meridian International Research, “Ssbj ii airline and fractional markets,” 2001. Aviation House, Warwick, England.

[108] Messac, A., “From dubious construction of objective functions to the application of physical programming,” *AIAA Journal*, vol. 38, no. 1, 2000.

[109] Middleton, W. and Lundry, L., “A system of aerodynamic design and analysis of supersonic aircraft,” Tech. Rep. CR-3351, National Air and Space Administration, 1980.

[110] Moore, M., “Vehicle sketch pad documentation,” 2004.

[111] Morgenstern, J., "Nacelle integration with reflexed wing for sonic boom reduction," Feb 2005.

[112] Morrison, M., Norris, G., Sarsfield, K., Warwick, G., and Wastnage, J., "Las vegas witlesses birth of supersonic gamble," *Flight International*, vol. 166, Oct. 2004.

[113] Mosher, T., "Conceptual spacecraft design using a genetic algorithm trade selection process," *Journal of Aircraft*, no. 1, 1999.

[114] Nelder, J. and Mead, R., "A simplex method for function minimization," *The Computer Journal*, vol. 7, no. 3, pp. 308–313, 1965.

[115] Nishino, H., Takagi, H., Cho, S.-B., and Utsumiya, K., "A 3d modeling system for creative design.," in *ICOIN*, pp. 479–486, 2001.

[116] Nowostawski, M. and Poli, R., "Parallel genetic algorithm taxonomy," 1999. KES'99.

[117] Ohsaki, M. and Takagi, H., "Improvement of presenting interface by predicting the evaluation order to reduce the burden of human interactive ec operators," in *Int'l Conf. on System, Man, Cybernetics*, 1998.

[118] Pace, S., "Northrop grumman unveils quiet, supersonic aircraft," *Flight Journal*, Feb. 2003.

[119] Parmee, I., "Genetic algorithms and hydropower system design," *Computer-Aided Civil and Infrastructure Engineering*, vol. 13, no. 1, 1998.

[120] Parmee, I., Cvetkovic, D., Watson, A., and Bonha, C., "Multi-Objective Satisfaction within an Interactive Evolutionary Design Environment," *Journal of Evolutionary Computation*, vol. 8, no. 2, 2000.

[121] Pawlowski, J., Graham, D., Boccadoro, C., Coen, P., and Maglieri, D., "Origins and overview of the shaped sonic boom demonstration program," Tech. Rep. AIAA-2005-5, 43rd AIAA Aerospace Sciences Meeting and Exhibit, Jan 2005.

[122] Phoenix Integration, "Modelcenter product information page," 2005. http://www.phoenix-int.com/products/ModelCenter.php.

[123] Plotkin, K. and Grandi, F., "Computer Models for Sonic Boom Analysis: PCBoom4, CABoom, BooMap, CORBoom," Tech. Rep. WR 02-11, Wyle Labs, 2002.

[124] Powell, M., "An efficient method for finding the minimum of a function of several variables without calculating derivatives," *The Computer Journal*, vol. 7, no. 3, pp. 155–162, 1964.

[125] Rallabhandi, S., *Sonic Boom Minimization through Vehicle Shape Optimization and Probabilistic Acoustic Propagation*. PhD thesis, Georgia Institute of Technology, 2005.

[126] Rasheed, K., "An adaptive penalty approach for constrained genetic-algorithm optimization," in *Genetic Programming 1998: Proceedings of the Third Annual Conference* (Koza, J. R., Banzhaf, W., Chellapilla, K., Deb, K., Dorigo, M., Fogel, D. B., Garzon, M. H., Goldberg, D. E., Iba, H., and Riolo, R., eds.), (University of Wisconsin, Madison, Wisconsin, USA), pp. 584–590, Morgan Kaufmann, 22-25 1998.

[127] Raymer, D., *Aircraft Design: A Conceptual Approach*. Washington, D.C.: American Institute of Aeronautics and Astronautics, 1992.

[128] Raytheon Aircraft Company, "Supersonic civil aircraft study," 2004. Presentation given at the 42nd AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV.

[129] Rentema, D., *AIDA: Artificial Intelligence supported conceptual Design of Aircraft*. PhD thesis, Delft University, 2004.

[130] Richtey, T., "Fritz zwicky, 'morphologie' and policy analysis," tech. rep., Presented at the 16th Euro Conference on Operational Analysis, 1998.

[131] Roskam, J., *Airplane Design*. Roskam Aviation and Engineering Corporation, 1990.

[132] Roth, G. and Crossley, W., "Commercial transport aircraft conceptual design using a genetic algorithm based approach," in *7th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, (Washington, D.C.), AIAA, 1998.

[133] Salas, A. O., Walsh, J. L., Mason, B. H., Weston, R. P., Townsend, J. C., Samareh, J. A., and Green, L. L., "Hsct4.0 application - software requirements specification," Tech. Rep. NASA/TM-2001-210867, National Air and Space Administration, May 2001.

[134] Schaffer, J. D., "Multiple objective optimization with vector evaluated genetic algorithms," in *Proceedings of the 1st International Conference on Genetic Algorithms*, pp. 93–100, Lawrence Erlbaum Associates, Inc., 1985.

[135] Schrage, D., *Concurrent Engineering: Automation, Tools, and Techniques*, ch. Concurrent Design: A Case Study. John Wiley and Sons, 1993.

[136] Seabass, R. and George, A., "Sonic boom minimization," *Journal of the Acoustical Society of America*, vol. 51, no. 2, 1972.

[137] Sexstone, M., "Aircraft structural mass property prediction using conceptual-level structural analysis," Tech. Rep. SAWE 2410, 57th conference of the SAWE, 1998.

[138] Shields, E. and Hicks, M., "Supersonic 150 passenger baseline configuration," 2003. Memo to Systems Analysis Branch from Swales Aerospace.

[139] Shields, W., Fenbert, J., Ozoroski, L., and Geiselhart, K., "Concept Development of a Mach 1.6 High-Speed Civil Transport," Tech. Rep. NASA/TP-1999-209697, Hampton, VA, 1999.

[140] Simmons, F. and Freund, D., "Morphing concept for quiet supersonic jet boom mitigation," Tech. Rep. AIAA-2005-1015, 43rd AIAA Aerospace Sciences Meeting and Exhibit, Jan. 2005.

[141] Sobieszczanski-Sobieski, J., "Optimization by decomposition: A step from hierarchic to non-hierarchic systems," in *2nd NASA/USAF Symposium on Recent Advances in Multidisciplinary Analysis and Optimization*, (Hampton, VA), NASA, 1988.

[142] Sobieszczanski-Sobieski, J., Agte, J., and Jr., R. S., "Bi-Level Integrated System Synthesis (BLISS)," in *7th AIAA/USAF/NASA/iSSMO Symposium on Multidisciplinary Analysis and Optimization*, (St. Louis, MO), AIAA, 1998.

[143] Spears, W. M., *The Role of Mutation and Recombination in Evolutionary Algorithms*. PhD thesis, The George Mason University, 1998.

[144] Srinivas, N. and Deb, K., "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1994.

[145] Stevens, S., "Perceived level of noise by mark vii and decibels," *J. Acoust. Soc. America*, vol. 51, Feb 1972.

[146] Stone et. al., "Evaluation of Equivalent Laminated Plate (ELAPS) Solution in HSCT Sizing," 2000. SAWE 3003.

[147] Sweetman, B., "Whooshhh!," *Popular Science*, July 2004.

[148] T. Hiroyasu, M. M. and Y.Tanimura, "The differences of parallel efficiency between the two models of parallelgenetic algorithms on pc cluster systems," in *Proceedings of the fourth international conference/exhibition of highperformance computing in Asia Pacific region*, 2000.

[149] Takagi, H., "Interactive evolutionary computation: Fusion of the capacities of ec optimization and human evaluation," *Proceedings of the IEEE*, vol. 89, no. 9, 2001.

[150] Takagi, H. and Ohya, K., "Discrete fitness values for improving the human interface in interactive ga," in *ICEC'96*, (Aichi, Japan), IEEE, 1996.

[151] Teal Group Corporation, "Small supersonic vehicle definition and market outlook," 2002. Teal Group.

[152] The Mathworks Inc., "Matlab - the language of technical computing," 2005. http://www.mathworks.com.

[153] Thomas, C., "Extrapolation of sonic boom pressure signatures by the waveform parameter method," Tech. Rep. TN D-6832, National Air and Space Administration, 1972.

[154] Thompson, A., Layzell, P., and Zebulum, R., "Explorations in design space: Unconventional electronics design through artifcial evolution," *IEEE Trans. Evol. Comp*, no. 3(3), 1999.

[155] Tong, S. S., Powell, D., and Goel, S., "Integration of artificial intelligence and numerical optimization techniques for the design of complex aerospace systems," Tech. Rep. AIAA-92-1189, Aerospace Design Conference, Feb. 1992.

[156] Wallace, D., Hayes, R., and Kulsrud, H., "Sonic boom propogation in a stratified atmosphere with computer program," Tech. Rep. CR-1299, Aeronautical research associates of Princeton, Inc., 1969.

[157] Whitford, R., *Design for Air Combat.* Alexandria, VA: Jane's Information Group, 1987.

[158] Whitham, G., "The flow pattern of a supersonic projectile," *Commun. Pure Appl. Math*, vol. 5, 1952.

[159] Whitley, D., "A genetic algorithm tutorial," *Statistics and Computing*, vol. 4, 1994.

[160] Wlezien, R. and Veitch, L., "The darpa quiet supersonic platform program," Tech. Rep. 2002-0143, AIAA, 2002.

[161] Wolz, R., "A summary of recent supersonic vehicle studies at gulfstream aerospace," Tech. Rep. AIAA-2003-558, American Instiitute of Aeronautics and Astronautics, Jan. 2003.

[162] Wolz, R., "A summary of recent supersonic vehicle studies at gulfstream aerospace," Tech. Rep. AIAA-2003-0558, 41st AIAA Aerospace Sciences Meeting and Exhibit, Jan. 2003.

[163] Wood, K. D., *Aerospace Vehicle Design Vol. I, Aircraft Design.* Boulder, CO: Johnson Publishing, 1963.

[164] Zitzler, E., Laumanns, M., and Thiele, L., "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," Tech. Rep. 103, Gloriastrasse 35, CH-8092 Zurich, Switzerland, 2001.

[165] Zitzler, E. and Thiele, L., "An evolutionary algorithm for multiobjective optimization: The strength pareto approach," Tech. Rep. 43, Gloriastrasse 35, CH-8092 Zurich, Switzerland, 1998.

[166] Zorumski, W. E., "Aircraft noise prediction program theoretical manual, parts 1 and 2," Tech. Rep. TM-83199, National Air and Space Administration, Feb. 1982.

[167] Zoutendijk, G., "Methods of feasible directions," 1960. Elsevier, Amsterdam.

[168] Zwicky, F., "Morphology of aerial propulsion," *Helvetica Physica Acta*, vol. XXI, no. 5, pp. 299–340, 1948.

[169] Zwicky, F., *Discovery, Invention, Research - Through the Morphological Approach.* Toronto: The Macmillian Company, 1969.

VITA

Michael Alexander Buonanno was born in Troy, NY in 1979. He graduated from La Salle Institute in 1997, and moved to Chicago, IL to attend the Illinois Institute of Technology. After receiving his Bachelors degree in Aerospace Engineering in 2001, he joined the Aerospace Systems Design Lab at Georgia Tech to pursue his Ph.D. studies.